

Computational Modeling Methods for Neuroscientists



edited by
Erik De Schutter



Computational Modeling Methods for Neuroscientists

Computational Neuroscience

Terence J. Sejnowski and Tomaso A. Poggio, editors

The Computational Brain, P. S. Churchland and T. J. Sejnowski, 1992

Dynamic Biological Networks: The Stomatogastric Nervous System, R. M. Harris-Warrick, E. Marder, A. I. Selverston, and M. Moulins, eds., 1992

The Neurobiology of Neural Networks, D. Gardner, ed., 1993

Large-Scale Neuronal Theories of the Brain, C. Koch and J. L. Davis, eds., 1994

The Theoretical Foundations of Dendritic Function: Selected Papers of Wilfrid Rall with Commentaries, I. Segev, J. Rinzel, and G. M. Shepherd, eds., 1995

Models of Information Processing in the Basal Ganglia, J. C. Houk, J. L. Davis, and D. G. Beiser, eds., 1995

Spikes: Exploring the Neural Code, F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, 1997

Neurons, Networks, and Motor Behavior, P. S. Stein, S. Grillner, A. I. Selverston, and D. G. Stuart, eds., 1997

Methods in Neuronal Modeling: From Ions to Networks, second edition, C. Koch and I. Segev, eds., 1998

Fundamentals of Neural Network Modeling: Neuropsychology and Cognitive Neuroscience, R. W. Parks, D. S. Levine, and D. L. Long, eds., 1998

Fast Oscillations in Cortical Circuits, R. D. Traub, J. G. R. Jeffreys, and M. A. Whittington, 1999

Computational Vision: Information Processing in Perception and Visual Behavior, H. A. Mallot, 2000

Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems, C. Eliasmith and C. H. Anderson, 2003

The Computational Neurobiology of Reaching and Pointing, R. Shadmehr and S. P. Wise, eds., 2005

Dynamical Systems in Neuroscience, E. M. Izhikevich, 2006

Bayesian Brain: Probabilistic Approaches to Neural Coding, K. Doya, S. Ishii, A. Pouget, and R. P. N. Rao, eds., 2007

Computational Modeling Methods for Neuroscientists, E. De Schutter, ed., 2010

For a complete list of books in this series, see http://mitpress.mit.edu/Computational_Neuroscience

Computational Modeling Methods for Neuroscientists

edited by Erik De Schutter

**The MIT Press
Cambridge, Massachusetts
London, England**

© 2010 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

For information about special quantity discounts, please email special_sales@mitpress.mit.edu

This book was set in Times New Roman on 3B2 by Asco Typesetters, Hong Kong.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Computational modeling methods for neuroscientists / edited by Erik De Schutter.

p. cm.—(Computational neuroscience series)

Includes bibliographical references and index.

ISBN 978-0-262-01327-7 (hardcover : alk. paper) 1. Computational neuroscience. 2. Neurobiology—
Mathematical models. I. De Schutter, Erik. II. Series: Computational neuroscience.

[DNLM: 1. Models, Neurological. 2. Neurosciences—methods. WL 20 C738 2010]

QP357.5.C625 2010

612.8—dc22

2009006125

10 9 8 7 6 5 4 3 2 1

Contents

Series Foreword	vii
Introduction	ix
1 Differential Equations	1
Bard Ermentrout and John Rinzel	
2 Parameter Searching	31
Pablo Achard, Werner Van Geit, and Gwendal LeMasson	
3 Reaction-Diffusion Modeling	61
Upinder S. Bhalla and Stefan Wils	
4 Modeling Intracellular Calcium Dynamics	93
Erik De Schutter	
5 Modeling Voltage-Dependent Channels	107
Alain Destexhe and John R. Huguenard	
6 Modeling Synapses	139
Arnd Roth and Mark C. W. van Rossum	
7 Modeling Point Neurons: From Hodgkin-Huxley to Integrate-and-Fire	161
Nicolas Brunel	
8 Reconstruction of Neuronal Morphology	187
Gwen Jacobs, Brenda Claiborne, and Kristen Harris	
9 An Approach to Capturing Neuron Morphological Diversity	211
Haroon Anwar, Imad Riachi, Sean Hill, Felix Schürmann, and Henry Markram	
10 Passive Cable Modeling	233
William R. Holmes	

11	Modeling Complex Neurons	259
	Erik De Schutter and Werner Van Geit	
12	Realistic Modeling of Small Neuronal Networks	285
	Ronald L. Calabrese and Astrid A. Prinz	
13	Large-Scale Network Simulations in Systems Neuroscience	317
	Reinoud Maex, Michiel Berends, and Hugo Cornelis	
	Software Appendix	355
	References	367
	Contributors	405
	Index	409

Series Foreword

Computational neuroscience is an approach to understanding the information content of neural signals by modeling the nervous system at many different structural scales, including biophysical, circuit, and system levels. Computer simulations of neurons and networks are complementary to traditional techniques in neuroscience. This book series welcomes contributions that link theoretical studies with experimental approaches to understanding information processing in the nervous system. Areas and topics of particular interest include biophysical mechanisms for computation in neurons, computer simulations of neural circuits, models of learning, representation of sensory information in neural networks, systems models of sensorimotor integration, and computational analysis of problems in biological sensing, motor control, and perception.

Terrence J. Sejnowski
Tomaso Poggio

Introduction

I am writing this introduction a week after Wilfrid Rall received the inaugural Swartz Prize for Theoretical and Computational Neuroscience. This event at the 2008 Society for Neuroscience meeting was a good demonstration of how much the field of computational neuroscience has moved into the mainstream. Compare this with the situation in 1989 when the first book in this MIT Press Computational Neuroscience series was published: *Methods in Neuronal Modeling: From Synapses to Networks*, edited by Christof Koch and Idan Segev. The first chapter of that seminal book on methods explained what computational neuroscience was about. Less than ten years later, in the second edition, such an introduction was no longer considered necessary. The present book takes the next logical step and introduces modeling methods to a broad range of neuroscientists.

The focus of this book is on data-driven modeling, i.e., the use of fairly standardized modeling methods to replicate the behavior of neural systems at different levels of detail. In general this will require numerical simulation of the model on a computer. In this aspect the book clearly differs from more theoretical approaches that study how the brain computes and processes information. An excellent introduction to that field is *Theoretical Neuroscience* by Peter Dayan and Larry F. Abbott, published in the same MIT Press series. Together, this book and the Dayan and Abbott one give a fairly complete overview of the current state of the field of computational neuroscience. Both books assume a basic knowledge of neuroscience in order to understand the examples given and are therefore more suited for neuroscientists and biologists than for scientists with a theoretical training entering the field. The latter are advised to first study a basic neuroscience textbook. Scientists with a biological or related background will also appreciate that this book tries to keep the required mathematics at an introductory level and in addition starts with a chapter that describes the necessary basic mathematical knowledge.

Data-driven modeling is a concept widely used in systems biology and covers the numerical methods described in this book. The models are based on a limited set of possible equations and the difference between them is mainly in the parameters used,

which are directly derived from experimental data. The model output is also in units that can be easily related and compared with experimental measurements. Although sometimes criticized as being underconstrained—“with so many parameters one can make the model do anything”—in practice the opposite is often observed: it can be quite difficult to develop a complex model that faithfully reproduces a wide range of phenomena. These “realistic” models can then be put to several uses.

The first and simplest application is as a formal quantification of the description of a neural system. Here experimentalists use a model to formalize their intuition on the dynamic properties of a system as derived from a series of experiments. This frequently results in a small modeling section at the end of a publication. Although these *demonstration models* often provide little new insights compared with the accompanying experiments, they are nevertheless an important sanity check. By demonstrating that the numbers really add up, they increase the validity of the data interpretation compared with the more qualitative conclusions in pure experimental papers. An example of fitting simple models to experimental data can be found in section 7 of chapter 5. It is rewarding to notice that there has been a sharp increase in combining this type of modeling with experimental work over past years. This trend is facilitated by the availability of a mature set of simulation software programs (see the software appendix at the end of this volume), the easy access to computational modeling courses, and the increasing online availability of model scripts. We hope that this book will be of good use to future practitioners of this approach.

The next step is to make modeling an explicit part of the research design itself. One approach is to have a *reference model* of the neural system being studied that is improved incrementally. This is accomplished as a cycle in which experimental data are used to develop a model and the model is then exploited to design new experiments, leading to corrections or expansions of the model. The model can inspire experiments in two different ways. The first is the identification of missing data. The story has been repeated many times: an experienced laboratory starts a modeling project convinced that it has built up a complete description of the neural system and then discovers that a number of crucial measurements are absent. The model functions here as a synthesis of the knowledge about a system and allows the developer to identify lacunas. By allowing investigation of all its variables simultaneously, it also provides a much richer description of the system’s dynamics than anything that can be put on paper. Once the lacunas are filled, one can move to the next stage: using the reference model to test quantitative understanding of the system. Here simulations are used to check whether the model faithfully reproduces important characteristics, including “emergent” properties for which the model was not fitted. The failure to do so is not seen as negative but instead provides the impetus to try to understand what is wrong or missing in the model; usually a new series of experiments is required before improvements can be made. Section 2 in chapter 12 provides a nice example of this approach in invertebrate neurophysiology. Although this paradigm

clearly requires an understanding of both experimental and modeling methods, in practice it is not easy for a single scientist to combine both. The problem is not only the training and experience needed, but also the time required to execute both. A common misconception is that a modeling project can be achieved quickly, which is almost never the case. Therefore the experiment–modeling cycle is best implemented as a team effort with close interaction of all partners.

The final application is *predictive modeling*, which is a purely theoretical application of the model. This does not require data-driven models per se, but they can be quite useful. Predictive data-driven modeling has been the focus of my own work and has led to several interesting results, briefly described in chapter 11, section 3 and chapter 13, section 3. The initial approach is similar to the one just described in that a reference model based on extensive experimental data is developed. This is then used to produce predictions of system properties for which no data are available. This is typically the work of theoreticians who invest a lot of time and effort in creating the model and then use it to do “experiments in silico.” Although it is not always appreciated by experimentalists, this approach will grow increasingly important. As experimental studies of neural systems become detailed and more complex, the resulting knowledge often becomes fragmented and lacking in integration. Theoreticians are more interested in building a “complete” model using data from many different sources, which is very time-consuming work. Therefore the reference model will be based on larger datasets, resulting in more complex representations that can generate richer behaviors. The predictions made can sometimes be tested experimentally and may therefore inspire work by other labs, but in many cases the tools to do this properly simply do not exist (yet). In this case, experimentation in silico is the only approach possible for investigating the properties of interest. An example is the Blue Brain Project (chapter 9), which is building a morphologically and physiologically exact model of a complete cortical microcolumn to study its function in great detail. The resulting model allows virtual patch clamping of all neurons in a microcolumn simultaneously.

A final stage of application can be model reduction; a more complete knowledge of the system allows its salient properties to be captured by one or a few equations. It is sometimes argued that understanding is not complete until this stage is achieved. Such approaches are common in theoretical neuroscience and are largely outside the scope of this book, although some examples can be found in chapter 1 and section 7 of chapter 13.

Before a model can be applied it needs to be built, and it is at this stage that this methods book will be most useful. The chapters are structured in intuitive order and try to cover all aspects of neural modeling, from molecules to networks. The authors have different backgrounds; some were trained as biologists or medical doctors, others have physics or mathematical backgrounds. Nevertheless, all share an active interest in realistic modeling in close interaction with experimental work, either in

their own labs or through collaboration. The content of the book is closely integrated, but each chapter has its own flavor, and many authors clearly state their preferred approach to modeling, with subtle differences among each other. This reflects the richness of computational neuroscience, a field that has attracted scientists with diverse training.

We start with two introductory chapters that provide the basic tools needed for all other chapters: an introduction to the mathematics of equation solving (chapter 1) and automated parameter-fitting methods (chapter 2). The latter has become an increasingly important technique in neural modeling because even for data-driven models, in neuroscience quantitative, trustworthy data to determine many of the parameters needed do not exist. In this aspect, computational neuroscience is quite distinct from systems biology, where data are usually much more complete (De Schutter, 2008). Next follow eleven chapters that increase in spatial scale of a model. Chapters 3 and 4 concern modeling of molecular reactions, which is useful to simulate signaling pathways and calcium dynamics. Next comes modeling of channels, voltage-gated ion channels (chapter 5) and synaptic ligand-gated ones (chapter 6); chapter 6 also covers learning rules. The majority of chapters concern modeling of single neurons. We start with point neurons (chapter 7), i.e., neurons without shape that form the basis of many network models. Next come neuron models with dendrites. Here we cover the methods needed to reconstruct (chapter 8) and correct (chapter 9) morphologies and then continue with passive (chapter 10) and active (chapter 11) models of neurons with complex morphologies. The final two chapters cover network modeling: realistic modeling of small microcircuits (chapter 12) and large networks in systems neuroscience (chapter 13). Every chapter covers the equations needed to simulate these models, sources of data for the model parameters, approaches to validate the models, and a short review of relevant models.

The authors and editor have tried to integrate these chapters into a coherent work. I wish to thank the authors and several colleagues for help in reviewing the comprehensive content of this book. We have minimized overlap among different chapters and provided extensive cross-referencing. The latter, together with the index, should allow readers to focus on only one or a few chapters that cover their area of interest, using the cross-references to look up additional material they are not familiar with elsewhere in the book. An important resource is the software appendix. This contains a short description and important information on the simulation software packages described in the book.

The book is complemented by a website at <http://www.compneuro.org>, which provides three services. It contains a list, chapter by chapter, of all urls mentioned in the book so that the reader can just click on them. It will also supply corrections to the book if needed and updates to the software appendix because this information evolves rapidly.

1 Differential Equations

Bard Ermentrout and John Rinzel

1.1 What Is a Differential Equation?

All processes in neuroscience change over time and it is thus necessary to model these processes as dynamic systems, that is, systems whose current and future states can be determined from their past. The most common and simplest types of dynamic models come in the form of differential equations in which the rate of change of some variable, say $u(t)$, is given as a function of u and other quantities. For example,

$$\frac{du}{dt} = -u/\tau + I(t). \quad (1.1)$$

This simple differential equation compactly embodies the rule for prediction of the system's entire future. It states that the instantaneous rate of change, increase or decrease, of $u(t)$ is the sum of a memory loss term with rate $-u(t)/\tau$ and an input rate $I(t)$. Many dynamic processes can be described by differential equations as instantaneous rules in this way:

$$\frac{du}{dt} = \text{change per unit time} = \text{sum of production rates} - \text{sum of removal rates}. \quad (1.2)$$

As a familiar example, suppose we are modeling the time course of a neuron's membrane potential $V(t)$. If c is the capacitance of the neuron's membrane, then the balance law for growth or decay of a charge on the membrane is

$$c \frac{dV}{dt} = I_{\text{ext}} - \sum I_{\text{inward}}(V, t) - \sum I_{\text{outward}}(V, t), \quad (1.3)$$

where we have grouped the different inward and outward currents that flow through the membrane's ionic channels. This is a differential equation for $V(t)$. It is called an *ordinary* differential equation (ODE) because there is only one independent variable: time, t .

More generally, we may combine the production and removal terms in a net input rate function $G(u, t)$. Here, by including u as an argument in G , we allow that G would generally have contributions that depend on $u(t)$ itself, and perhaps nonlinearly:

$$\frac{du}{dt} = G[u(t), t]. \quad (1.4)$$

A complete formulation for predicting the future, say, from some particular starting time t_0 requires that we also specify the state at $t = t_0$, $u(t_0)$. Prediction corresponds to solving the differential equation given the initial condition.

How do we solve the differential equation (1.4) for $u(t)$? How do we obtain the prediction for the future of u ? One can imagine doing this computationally by marching forward in time, taking small steps in time of size Δt . That is, we make the approximation and replace the differentials dt and du by differences Δt and $\Delta u = u(t + \Delta t) - u(t)$. Then equation (1.4) becomes

$$u(t + \Delta t) = u(t) + \Delta t G[u(t), t]. \quad (1.5)$$

This straightforward recipe is easy to implement and is guaranteed to work if Δt is small enough and if the differential equation is reasonable (see Mascagni and Sherman, 1998). That is, speaking more rigorously, under reasonable assumptions about the function $G(u, t)$ —namely, the derivative of G with respect to u is continuous and G is a continuous function of time—this equation guarantees that there exists a unique solution to the differential equation (1.4) for all times t close enough to t_0 .

Why do we care about this mathematical arcana? Since most differential equations that are written for neurons do not have a nice closed-form solution [that is, we cannot write an expression for the solution in terms of known functions like $\sin(t)$], we must generally use a computer to find solutions. The point of existence and uniqueness is that they tell you that there is a solution which, with care, can be well approximated by the computer. Later on we will see that there are some types of differential equations that do not have any solutions unless the parameters are carefully chosen. (Some ODE problems specify more conditions than there are ODEs and so will have solutions only for special values of the parameters. We illustrate this in the section on partial differential equations later in this chapter with the example of finding the speed and profile of a propagating impulse for a Hodgkin-Huxley-like cable equation.)

The formulation for a single unknown quantity, u , can be generalized to systems of differential equations, which we can write as

$$\begin{aligned}\frac{dx_1}{dt} &= G_1(x_1, \dots, x_n, t) \\ \frac{dx_2}{dt} &= G_2(x_1, \dots, x_n, t) \\ &\dots = \dots \\ \frac{dx_n}{dt} &= G_n(x_1, \dots, x_n, t).\end{aligned}\tag{1.6}$$

These equations are all evaluated in parallel. That is, given the values of all the n variables at the current time t , their values at $t + \Delta t$ are approximately

$$x_j(t + \Delta t) \approx x_j(t) + \Delta t G_j.\tag{1.7}$$

That is, the functions G_j specify the rates of change of each of the variables x_j and how they depend on the present values of those variables. We call an equation like (1.6) a *system of first-order differential equations*. For example, if the $x_j(t)$ terms represent, say, concentrations of chemicals involved in a set of chemical reactions, then the functions G_j are the rates of change for each chemical species and depend on the specific reactions and their reaction rates. In neuroscience applications, the variables x_j usually are the potentials of neurons, the fractions of open channels, and concentrations of ionic species. We remark that the time scales of all these interacting species can vary over many orders of magnitude. Yet, once their values are specified at a single point in time, we can know them for any other points in time for which they are defined.

Many physical phenomena involve more than just the rates of change. For example, a mass m attached to a spring with a spring constant k (ignoring friction) obeys Newton's equation:

$$\text{mass} \times \text{acceleration} = \text{force},\tag{1.8}$$

which in terms of differential equations is

$$m \frac{d^2x}{dt^2} = -kx.\tag{1.9}$$

Here the displacement of the mass (relative to rest), $x(t)$, is determined by its second derivative rather than just its rate of change. This is a *second-order* differential equation, and in order to specify $x(t)$ we need to know both the position, $x(t_0)$, and the velocity, $dx/dt(t_0)$, at the same time t_0 . If we introduce a new variable, the velocity, defined as $q(t) = dx/dt$, then we can write the equations for the mass-spring system as

$$\begin{aligned}\frac{dx}{dt} &= q \\ \frac{dq}{dt} &= -(k/m)x.\end{aligned}\tag{1.10}$$

Here we use the fact that $dq/dt = d^2x/dt^2$. We have reduced the second-order equation to a pair of first-order equations. In many cases of interest, systems of n th-order differential equations can be transformed into systems of first-order equations by introducing intermediate variables. The converse is not true: we cannot write most systems of first-order equations as a single n th-order system. For this reason, we will stick with systems of first-order differential equations in this chapter. (Most simulators of differential equations require that you write them as systems of first-order equations, so it is good to practice this.)

If the functions G_1, \dots, G_n are independent of the variable t , we call the system of equations *autonomous*. Otherwise this is a nonautonomous differential equation. In general, autonomous equations are easier to analyze. The behavior of solutions to ODEs can be as simple as convergence to a constant in time, or they can vary in an oscillatory fashion or even become chaotic.

Another class of models consists of *delay-differential* equations, in which the evolution rule depends not only on the current state but also on the system's state some finite time in the past. This might be useful, say, as a simplified way of accounting for propagation or synaptic delays between neurons in a network model. A simple example is the delayed negative feedback model:

$$\frac{du}{dt} = -u(t) + \frac{a}{1 + bu(t - t_d)}.\tag{1.11}$$

While the formulation is simple-looking, such models can have quite complex behaviors. The solution at time t depends, not on just the present time, but rather on a whole interval of the past. In order to compute behavior at a time t_0 , the behavior of $u(t)$ must be known for all $t_0 - t_d \leq t \leq t_0$. The analysis and simulation of delay equations require special methods and numerical techniques beyond the scope of this chapter. For descriptions of some dynamic properties and analysis of delay equations in biology, see Macdonald (1989).

In this chapter we will highlight a few properties of differential equations and concepts for understanding them. If you wish more elaboration, but without overburdening mathematical detail, we recommend the books by Edelstein-Keshet (2005), Izhikevich (2006), and Strogatz (2001), and the appendices in Fall et al. (2005).

1.2 Linear Systems

If the functions G_j are linear with respect to the variables, x_j , then we call the system *linear*. For example, equation (1.1) is a one-variable (or first-order) linear differential equation. If $I(t)$ is a constant, \bar{I} , the solution is a simple exponential:

$$u(t) = \bar{I}\tau + (u_0 - \bar{I}\tau) \exp[-(t - t_0)/\tau], \quad (1.12)$$

where $u_0 = u(t_0)$. Linear systems arise naturally as descriptions when nonlinearities are weak and negligible and, more generally, for considerations of stability and sensitivity of nonlinear systems in the neighborhood of an equilibrium operating point.

In the general case of n -variables it is convenient to write the system (1.6) in vector-matrix form:

$$\frac{d\mathbf{X}}{dt} = \mathbf{A}(t)\mathbf{X}(t) + \mathbf{f}(t). \quad (1.13)$$

Here $\mathbf{A}(t)$ is an $n \times n$ matrix, $\mathbf{X}(t)$ is the vector of variables (x_1, \dots, x_n) , and $\mathbf{f}(t)$ is the n -component vector of t -dependent input functions. For example, for the spring-mass system (1.10) in vector-matrix form, we would have the vector $\mathbf{X} = (x, q)$ and the matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -k/m & 0 \end{pmatrix}. \quad (1.14)$$

If the matrix entries and the function, $\mathbf{f}(t)$, are continuous with respect to time, then there exists a unique solution to the linear equation with $\mathbf{X}(t_0) = \mathbf{X}_0$ for all time. This is much better than we can hope for in the nonlinear case. [The reader should verify that the solution to $dx/dt = x^2$, with $x(0) = 1$ is $1/(1-t)$, which exists only for $t < 1$.] We say that a linear system is *homogeneous* if $\mathbf{f}(t) = \mathbf{0}$. Linear homogeneous autonomous equations are the only general class of equations for which an explicit solution can always be written. If the entries of \mathbf{A} are constant in time, the solution to

$$\frac{d\mathbf{X}}{dt} = \mathbf{AX} \quad (1.15)$$

is a sum of n exponentials:

$$\mathbf{X}(t) = c_1 \mathbf{V}_1 e^{\lambda_1 t} + \cdots + c_n \mathbf{V}_n e^{\lambda_n t}. \quad (1.16)$$

The c_j are arbitrary constants and the pairs \mathbf{V}_j, λ_j comprise the n “modes” of the system, determined solely by the matrix \mathbf{A} . Since each exponential term of this sum must satisfy (1.15), we have (by substituting $\mathbf{V}_j e^{\lambda_j t}$ into equation 1.15) that

$$\lambda_j V_j = \mathbf{A} V_j. \quad (1.17)$$

These scalar-vector pairs are called the eigenvalue-eigenvector pairs for the matrix \mathbf{A} .

The eigenvalues, λ_j , are the roots to the n th degree polynomial,

$$p_A(\lambda) \equiv \det(\lambda\mathbf{I} - \mathbf{A}). \quad (1.18)$$

If the matrix \mathbf{A} is real-valued, and there is a complex eigenvalue, $\lambda = r + i\omega$, with an eigenvector, $V = \mathbf{U} + i\mathbf{W}$, then there is also a complex conjugate eigenvalue $\bar{\lambda} = r - i\omega$ and a complex conjugate eigenvector $\bar{V} = \mathbf{U} - i\mathbf{W}$, where \mathbf{U} , \mathbf{W} , r , ω are all real. This means that we can choose the constants in equation (1.16) so that the resulting solutions to the differential equation are all real. We remind the reader of the important identity:

$$e^{(r+i\omega)t} = e^{rt}(\cos \omega t + i \sin \omega t). \quad (1.19)$$

For example, referring back to our spring-mass system, the characteristic equation for the eigenvalues is

$$0 = \det(\lambda\mathbf{I} - \mathbf{A}) = \lambda^2 + m/k. \quad (1.20)$$

Therefore the eigenvalues are purely imaginary: $\lambda_{1,2} = \pm i\omega$ where $\omega = (k/m)^{1/2}$. The mass oscillates with a frequency ω .

Now let's discuss an important aspect of linear autonomous differential equations: notions of stability. First note that $X = 0$ is always a solution to (1.15). That is, if we start with nothing we stay with nothing. A solution that does not change with time is called an *equilibrium point* or *fixed point* of the differential equation. If we choose initial conditions close to the equilibrium point $X = 0$, it is natural to ask if the solution will stay close and even if it will decay back to the equilibrium. Any solution to the linear differential equation can be written in terms of the general solution $X(t)$ given by (1.16). Write $\lambda_j = r_j + i\omega_j$, where if the eigenvalue is real, $\omega_j = 0$. If each of the r_j is negative, then all solutions to the linear equation exponentially decay to zero. In this case, we say zero is *linearly asymptotically stable*. If at least one of the r_j is positive, then some solutions to equation (1.15) will grow exponentially and zero will be linearly unstable. Suppose that all the $r_j < 0$ except for one pair of imaginary eigenvalues, $\pm i\omega$. Then all components of $X(t)$ will decay except for those corresponding to the imaginary eigenvalues, and we are left with a sinusoidally varying solution $X(t)$ with a period of $2\pi/\omega$. We say that zero is *neutrally stable* in this case. Neutral stability is special since small changes in the coefficients of \mathbf{A} will destroy the neutral stability and make zero either asymptotically stable or unstable.

It is useful to find conditions on the matrix \mathbf{A} such that solutions are stable, unstable, or neutrally stable. This is a difficult problem, but it can be addressed by looking at the characteristic polynomial, $p_A(\lambda)$. Asymptotic stability is guaranteed if all

the roots of the characteristic polynomial $r_j + i\omega_j$ satisfy $r_j < 0$. If we write this polynomial as

$$p_A(x) = x^n + q_{n-1}x^{n-1} + \cdots + q_1x + q_0, \quad (1.21)$$

then it turns out that there is a simple set of conditions on the coefficients q_k that guarantee an asymptotic stability called the Routh-Hurwitz criterion. This can be found in many textbooks (e.g., Edelstein-Keshet, 2005, p. 233), so we do not state it here. Rather, we write the conditions for $n = 1, 2, 3, 4$. The state 0 is asymptotically stable if and only if

$$n = 1. \quad q_0 > 0.$$

$$n = 2. \quad q_0 > 0, q_1 > 0.$$

$$n = 3. \quad q_2 > 0, q_0 > 0, q_1 q_2 > q_0.$$

$$n = 4. \quad q_3 > 0, q_1 > 0, q_0 > 0, q_3 q_2 q_1 > q_1^2 + q_3^3 q_0.$$

In the cases, $n = 2, 3, 4$, if the last inequality is an equality (replace the $>$ by an $=$), then there is neutral stability and exactly one pair of imaginary eigenvalues, $\pm i\omega$.

It is instructive to look at the $n = 2$ case in more detail because we will need it later in this chapter. The matrix A has the form

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (1.22)$$

and the characteristic polynomial is

$$x^2 - (a + d)x + ad - bc. \quad (1.23)$$

The reader will recognize that $a + d$ is the trace (sum of the diagonals) of the matrix and $ad - bc$ is the determinant. Thus all solutions to a linear two-dimensional (2-D) autonomous system decay to zero exponentially if and only if the trace ($a + d$) is negative and the determinant ($ad - bc$) is positive. *These are not conditions for any higher-order system; only $n = 2$.*

In summary, linear (constant coefficient) autonomous systems have solutions that are sums of exponentials. The exponents are eigenvalues of the matrix A and their real parts determine the stability of the system about equilibrium.

1.3 Nonlinear Systems: Geometric Approach, Phase Planes

Although nonlinear systems can rarely be solved analytically in closed form, we have useful tools for characterizing their dynamic properties. These methods enable us to

find parameter regimes where behaviors of interest arise, e.g., where systems change from steady-state behavior to oscillatory behavior. Without getting into specialized approximations and lengthy analytic contortions and expressions, we can gain qualitative insight from a geometric viewpoint and from dynamic systems software. The sources (Strogatz, 2001; Izhikevich, 2006; Edelstein-Keshet, 2005) are particularly oriented toward the geometric approach.

Some Definitions and Preliminaries

We can write a general nonlinear ODE in vector form:

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}), \quad (1.24)$$

where \mathbf{X} is a vector of variables and \mathbf{F} is a vector of functions that specify the rates of change for the variables; $\mathbf{F}(\mathbf{X}) = [f_1(\mathbf{X}), \dots, f_n(\mathbf{X})]$. The vector, $\mathbf{X}(t) = [x_1(t), \dots, x_n(t)]$ is called the state vector. For our purposes, $x_j(t)$ are real and thus the vector \mathbf{X} lies in the n -dimensional real space, R^n . We call the space in which the variables lie the *state space* or the *phase space* of the system. For example, if there is only one variable, then the state space is the line and if there are two variables, the state space is the plane, or *phase plane*. Solution curves $\mathbf{X}(t)$ trace out curves in the phase space called *trajectories*. These are always one-dimensional curves parameterized by t , the time variable. One special type of trajectory is the constant one, $\mathbf{X}(t) = \mathbf{X}^*$ where $\mathbf{F}(\mathbf{X}^*) = 0$. This is the equilibrium point, steady state, or fixed point of the system. Linear systems usually have only one equilibrium point, but nonlinear systems can have many.

Another class of solutions found in neuroscience are periodic trajectories, $\mathbf{X}(t) = \mathbf{X}(t + T)$, where $T > 0$ is the period. If no other neighboring trajectories are periodic, then this trajectory is called a *limit cycle*. Limit cycles and other periodic trajectories require at least two dimensions in the phase space. Periodic trajectories (also called periodic orbits) trace out a closed curve in the phase space.

The idea of stability is a useful concept from differential equations that generalizes to nonlinear equations. A particular trajectory, $\mathbf{X}_p(t)$ (for example, a fixed point or limit cycle), is *stable* if any trajectories that are nearby at time t_0 remain nearby for all $t > t_0$. If any nearby trajectories diverge from $\mathbf{X}_p(t)$ as time increases, we say that $\mathbf{X}_p(t)$ is *unstable*. A stronger and, for neuroscience, more relevant form of stability requires in addition that as $t \rightarrow \infty$, nearby solutions converge to the orbit. For fixed points, this means that all nearby trajectories will tend to the fixed point for long times. For periodic orbits, nearby trajectories converge to the closed curve formed by the limit cycle. This type of stability is called *asymptotic stability*. The stability of equilibria is easy to determine by linearizing the equations about the equilibrium. That is, let \mathbf{X}^* be an equilibrium solution and write $\mathbf{X}(t) = \mathbf{X}^* + \mathbf{Y}(t)$. Then

$$\frac{d\mathbf{Y}}{dt} = \mathbf{F}(\mathbf{X}^* + \mathbf{Y}) \approx \mathbf{A}\mathbf{Y} + \mathbf{R}(\mathbf{Y}) \quad (1.25)$$

from Taylor's theorem. For a one-variable system, \mathbf{A} is the derivative $d\mathbf{F}/dx$ evaluated at \mathbf{X}^* ; for an n -th-order system, \mathbf{A} is an $n \times n$ matrix. $\mathbf{R}(\mathbf{Y})$ is the remainder term and contains quadratic and higher-order terms. If \mathbf{Y} is small, then the quadratic terms are much smaller than the linear terms. Since we are interested in what happens near the equilibrium point, we will assume that \mathbf{Y} is small. [Note that $\mathbf{Y}(t) = \mathbf{X}(t) - \mathbf{X}^*$ represents the deviation from the equilibrium point.] Truncating the remainder terms leaves us with

$$\frac{d\mathbf{Y}}{dt} = \mathbf{A}\mathbf{Y}, \quad (1.26)$$

the familiar equation (1.15). This linear system (or more commonly, the matrix \mathbf{A}) is the *linearization* of equation (1.24) around the equilibrium point \mathbf{X}^* . The matrix \mathbf{A} is the matrix of partial derivatives of \mathbf{F} evaluated at the equilibrium point; the entry in row i and in column j is the partial derivative $\partial f_i(\mathbf{X})/\partial x_j$, the derivative of f_i with respect to x_j , treating the other variables as parameters. Now, if all of the eigenvalues of \mathbf{A} have negative real parts, then $\mathbf{Y}(t)$ decays to zero and there is a theorem that says that the equilibrium point, \mathbf{X}^* , is asymptotically stable as a solution to the nonlinear system. Similarly, if any of the eigenvalues of \mathbf{A} have positive real parts, then the equilibrium point is unstable. There is an analogous linearization and stability theory for limit cycle solutions that provides a condition for a periodic orbit to be stable.

One-Dimensional Dynamics

We now turn to a geometric description of nonlinear systems—a viewpoint that is based on trajectories and structure of the phase space. Let's demonstrate with a simple example. Consider the nonlinear first-order system

$$\frac{dx}{dt} = f(x) + I, \quad (1.27)$$

where $f(x) = x(x - a)(1 - x)$, and suppose $0 < a < 1$. We start with the simple case in which $I = 0$. This nonlinear ODE has three different steady-state solutions: $x = 0, a, 1$. (This is what makes nonlinear systems much more interesting and useful; they can have many possible states.) Does this mean that the system it represents has three possible resting states? Potentially, yes; but in fact a steady state is, practically speaking, realizable only if it is stable. Applying the methods of the previous section, a steady state will be stable if the eigenvalues of the linearization have negative real parts. For a scalar model, the matrix \mathbf{A} is a scalar, $c = f'(x^*)$, where x^* is one of the

three steady states, 0, a , 1. Thus a condition for a steady state to be asymptotically stable is that $c = f'(x^*) < 0$. For our example, $f'(x) = -3x^2 + 2(a+1)x - a$, so that $f'(0) = -a < 0$ and $f'(1) = a-1 < 0$ since a is between 0 and 1. Thus both $x = 0$ and $x = 1$ represent stable steady states. For the case $x^* = a$ we find that this steady state is unstable since $f'(a) = a(1-a) > 0$; thus, we would not physically see this state. This simple system is bistable. Starting from an initial condition near $x = 0$ or $x = 1$, the system approaches one of the two stable states. If the parameter I is nonzero, then finding the equilibrium points is much harder (we have to solve a cubic equation) and it provides little insight. For this reason, we turn to the geometric approach.

The x-axis is the phase space for this one-variable dynamic system. Think of $x(t)$ as a point moving, as t increases, along the x-axis. Next, plot the rate function $f(x)$ on the vertical axis. Immediately, we know the direction in which $x(t)$ moves, and we indicate that flow with arrows on the x-axis (figure 1.1a). Where $f(x) > 0$, the arrow at x points rightward (since dx/dt is positive) and where $f(x) < 0$, the arrow is leftward. If $x(0)$ lies to the left of position $x = a$, then $x(t)$ moves smoothly toward $x = 0$, asymptotically approaching this stable steady state as $t \rightarrow \infty$. On the other hand, if $x(0)$ lies to the right of $x = a$, then $x(t)$ drifts toward $x = 1$. The two steady states $x^* = 0, 1$ are *attractors* and the phase space is divided into their two domains of attraction by the unstable steady state $x^* = a$. The bistable system shows a threshold behavior (figure 1.1b). It is rather elegant that the geometric, phase-space treatment gives us an immediate global view.

The preceding analytic approach was really only a local description, based on linearizing $f(x)$ around x^* . Although limited, it did give us the local rate of (exponential) approach or repulsion. The geometric approach shines brightest when we introduce the parameter I . As noted earlier, it is tedious to find the equilibrium points and their stability analytically for the cubic equation (1.27). However, with the geometric method, the addition of I does not change the methods and ideas. If $I > 0$, then the cubic picture in figure 1.1a is raised (see figure 1.1c) and if $I < 0$, it is lowered. Two interesting things happen as I is changed. As I increases, the two equilibria corresponding to 0, a merge when $I = I_2$ and no longer exist for $I > I_2$. The model has only one stable equilibrium point. Since the behavior is qualitatively different (monostable versus bistable), we say that a *bifurcation* takes place at $I = I_2$. If I is near I_2 , two distinct behaviors are observed, depending on whether I is above or below I_2 . As I is lowered, a bifurcation occurs at I_1 where the equilibria corresponding to 0, 1 merge and disappear. We summarize the behavior in the compact diagram shown in figure 1.1d, which is a bifurcation diagram. This diagram shows only the steady states and indicates if they are stable (solid) or unstable (dashed). For a fixed value of I between I_1 and I_2 , there are three steady states, while for $I > I_2$ or $I < I_1$ there is only one.

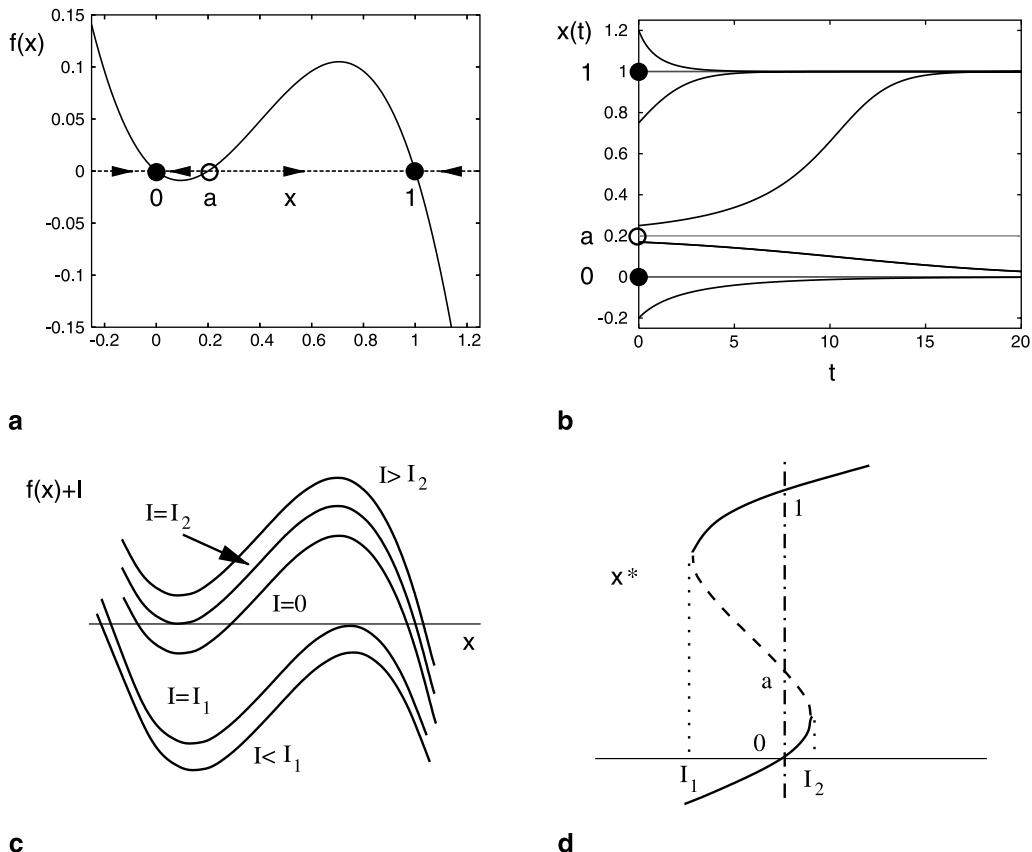


Figure 1.1
One-dimensional phase space.

Multidimensional Dynamics; Phase Planes

Clearly, the one-dimensional phase line is good only for one-variable systems. For higher-dimensional systems, the geometric methods are powerful but difficult to generalize except when the system is planar. Surprisingly, much of the behavior of neurons can be captured by looking at two-dimensional dynamics. So, let us jump into two dimensions and introduce phase-plane analysis. Consider the general two-variable system before we get specific:

$$\begin{aligned} \frac{dx}{dt} &= f(x, y) \\ \frac{dy}{dt} &= g(x, y). \end{aligned} \tag{1.28}$$

We interpret the pair of rates (f, g) geometrically. They specify the local velocity vector, the direction of motion at position (x, y) in the plane. At points where $f(x, y) = 0$, the x -direction velocity is zero. The points for which $f(x, y) = 0$ define a curve implicitly, a special curve in the plane, the x -nullcline. When a trajectory crosses the x -nullcline, the flow is vertical; $x(t)$ would be a local maximum or minimum. Similarly, the points where $g(x, y) = 0$ define the y -nullcline, along which the flow is horizontal. By drawing these two curves and then looking at a few local velocity vectors, we can learn a lot about the qualitative behavior of the 2-D dynamic system. Furthermore, at (x, y) points where the two nullclines intersect, both dx/dt and dy/dt vanish, so that these are steady states for the two-dimensional system. As with the one-dimensional system, we can immediately see the equilibrium points in the system (1.28) by graphing the nullclines and observing intersections. Unlike the one-dimensional system, the stability of the steady states cannot in general be determined by inspection. With interactive dynamical systems software packages such as XPPAUT (see the software appendix at the end of this volume) the flow field and nullclines appear at the push of a button.

Suppose that we have found a point $(x, y) = (x^*, y^*)$ that is an equilibrium or steady state: $f(x^*, y^*) = g(x^*, y^*) = 0$ (easily viewed in the phase plane by looking at the intersections of the nullclines.) Unlike the one-dimensional example, it is hard to discern stability from the phase plane alone so that we are forced to evaluate stability using the linearization techniques. We define

$$\mathbf{A} = \begin{pmatrix} \partial f(x^*, y^*)/\partial x & \partial f(x^*, y^*)/\partial y \\ \partial g(x^*, y^*)/\partial x & \partial g(x^*, y^*)/\partial y \end{pmatrix} \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (1.29)$$

to be the linearized system around an equilibrium point. From section (1.1) we see that the equilibrium will be stable if and only if $a + d < 0$ and $ad - bc > 0$.

As an example, consider the following model:

$$c \frac{dV}{dt} = I - g_L(V - E_L) - g_{Ca}m_\infty(V)(V - E_{Ca}) - g_Kw(V - E_K) \equiv cf(V, w) \quad (1.30)$$

$$\frac{dw}{dt} = [w_\infty(V) - w]/\tau_w(V) \equiv g(V, w), \quad (1.31)$$

which is a one-compartment model for the membrane potential of a barnacle muscle. [This so-called Morris-Lecar model (Morris and Lecar, 1981) is taken through its paces in Rinzel and Ermentrout (1998).] Here the x -axis will be the membrane potential, V , and the y -axis will be the fraction w of potassium channels that are open. (Note that the notation \equiv indicates that we are defining a symbol for an expression.)

In figure 1.2a we illustrate the geometric methods by showing the phase-plane portrait of equations (1.30)–(1.31); it is representative for models of many excitable

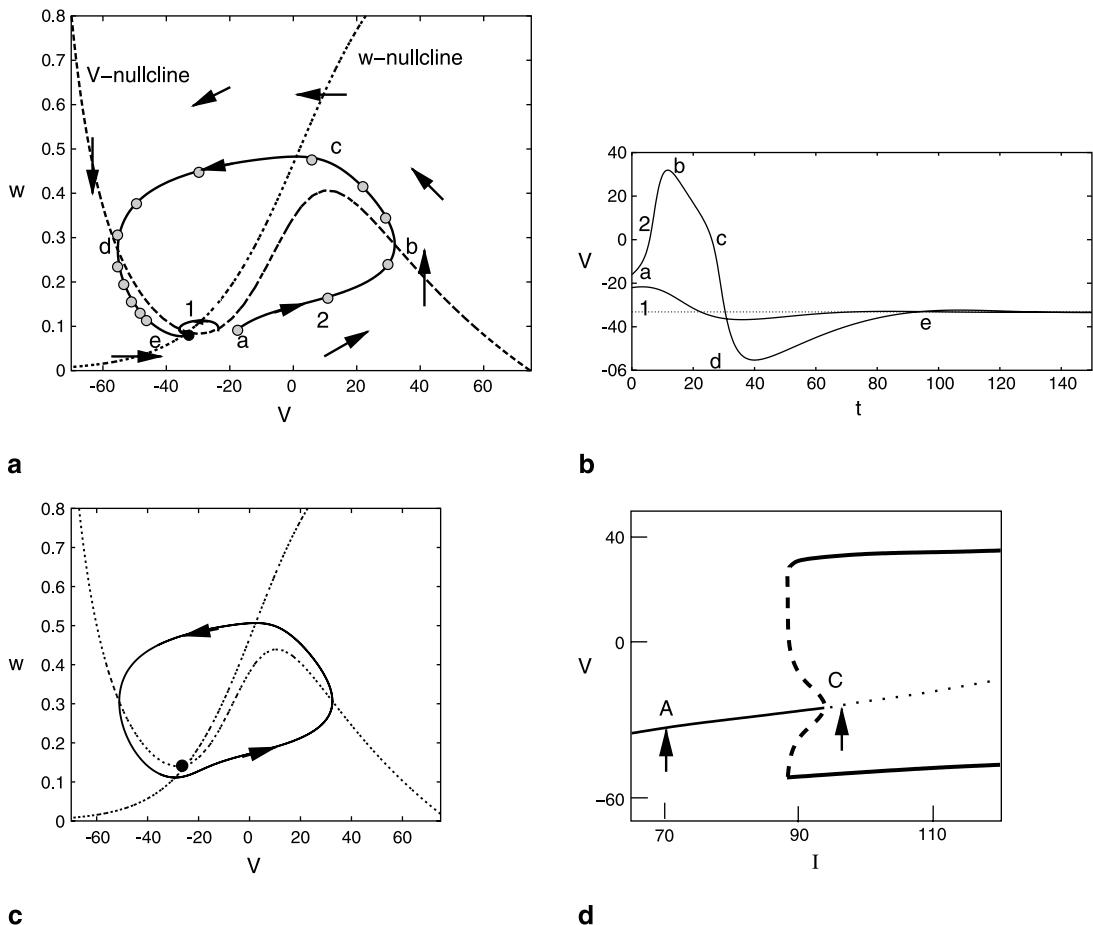


Figure 1.2
Two-dimensional phase space and excitability.

systems, such as two-variable models for an action potential and membrane electrical excitability. The point (V^*, w^*) where the two nullclines intersect corresponds to a steady state or fixed point of the system (shown as a filled circle). There, both dV/dt and dw/dt are zero. In this example, (V^*, w^*) is unique and is stable. We might refer to it as the resting state of the system.

Now let's look at the flow. Each point in the phase plane is associated with a direction found by evaluating $dV/dt = f(V, w)$ and $dw/dt = g(V, w)$ at the given point. If both dV/dt and dw/dt are positive at a point (V, w) , then they will be increasing at this point and we draw an arrow pointing up and to the right. In this model, dV/dt is positive for points below the V -nullcline and dw/dt is positive below

the w -nullcline. Hence, the nullclines partition the phase plane into regions where the flow is in one of four directions: up and to the right ($dV/dt > 0, dw/dt > 0$), up and to the left ($dV/dt < 0, dw/dt > 0$), down and to the right ($dV/dt > 0, dw/dt < 0$), and down and to the left ($dV/dt < 0, dw/dt < 0$). For points on the nullclines, either $dV/dt = 0$ or so the arrows are respectively vertical or horizontal.

Figure 1.2a shows some arrows indicating the directions at selected points as well as to trajectories (solutions) starting at different points in the plane. The velocity arrows show a generally counterclockwise flow. Starting from any initial position, the trajectory will eventually return the system back to the resting state. This is a powerful statement: the resting state (x^*, y^*) is a global attractor for the system. Moreover, from the layout of these nullclines and a few flow arrows, we see the essential features of excitability. A small perturbation from resting (say, in V) produces a response that does not grow in V ; the trajectory returns directly to rest (the trajectory labeled 1). If a large enough initial V -perturbation pushes the phase point rightward so that (at $t = 0^+$) it lies below the V -nullcline’s hump, then $dV/dt > 0$ and we get amplification and an action potential.

Figure 1.2b shows the voltage as a function of time for the subthreshold (1) and suprathreshold (2) initial conditions. For the suprathreshold initial conditions generating a spike, we label the parts of the action potential in figure 1.2b and their corresponding points in the phase plane in figure 1.2a. The suprathreshold trajectory takes off, heading toward the right branch of the V -nullcline. This demonstrates the thresholdlike behavior and the spike’s upstroke and occurs between points a and b. The trajectory turns vertical when it reaches the V -nullcline, corresponding to the peak voltage of the spike (at point b). Next, the flow carries the phase point to the upper left (point c), with w increasing and leftward drift constrained by the V -nullcline. Note that the trajectory cannot cross (leftward) back over the V -nullcline because of the rightward flow over there. This is the phase of growing negative feedback w (potassium current activation) during a spike’s depolarized (plateau) phase (between b and c). However, after the trajectory rises to above the V -nullcline’s upper “knee,” then it is free to dramatically turn leftward (dV/dt is strongly negative there). The spike’s downstroke proceeds, and maximum w occurs during the downstroke when the trajectory crosses the w -nullcline. With the fall of V to its minimum (point d), we return to rest during this recovery phase (point e).

The phase-plane plots do not provide information about the time it takes to go from one point to another: they are purely geometric concepts. However, one can plot a few gray circles as isolated points along the trajectory to give a sense of the timing between points. In figure 1.2a we add a few circles to illustrate the position every 5 ms for the first several points. During the upstroke and downstroke, the distance between successive points is larger than along the plateau and recovery. This type of behavior is typical in neural models where the voltage dynamics are typically much faster than the recovery.

The geometric view has enabled us to approximately predict the spike's trajectory without actually computing it. Many other properties of the system can be deduced from the phase-plane analysis, such as absolute and relative refractoriness, postinhibitory rebound, and repetitive firing (e.g., see Izhikevich, 2006, ch. 4). For example, the effect of a steady (depolarizing) stimulus current is to translate the V -nullcline upward (with a little distortion). As the stimulus increases, the equilibrium point moves toward the right (more depolarized) and upward (more recovery) along the w -nullcline (always, by definition, at the nullclines' intersection point). For an adequate stimulus, (V^*, w^*) will be on the middle branch of the V -nullcline (figure 1.2c). If w is relatively slow, then we can immediately conclude from the phase plane that (V^*, w^*) is unstable in this situation. The flow in the V -direction is strongly away from (x^*, y^*) ; it points leftward just to the left of (x^*, y^*) and rightward just to its right. In this case, trajectories converge onto the closed orbit of repetitive spiking; here, this *limit cycle* is the global attractor. Periodic solutions to the differential equation are represented by a closed trajectory in phase space. All solutions to the ODE (except for those starting at the unstable equilibrium point shown in the figure as a filled black dot) will converge to this closed trajectory.

Unlike the changes seen in figure 1.1, here the number of equilibria does not change as the parameter varies, but the stability of the equilibrium point *does* change. The change in stability of (V^*, w^*) can be characterized analytically. One linearizes about (V^*, w^*) and finds that for a critical stimulus level, the local behavior changes from a damped to a growing oscillation. The eigenvalues are a complex pair, $r \pm i\omega$, for which r changes from negative to positive at criticality. When an equilibrium point changes from stable to unstable via a pair of complex eigenvalues, then we expect to find limit cycles for values of the parameter near the transition point. This fact is a consequence of the *Hopf bifurcation theorem* (Edelstein-Keshet, 2005, p. 341).

Figure 1.2d shows the bifurcation diagram for equations (1.30–1.31) as the current I increases. The maximum and minimum values of the voltage are shown along the y-axis while the current is shown on the x-axis. Figure 1.2a corresponds to a current of $70 \mu\text{A}/\text{cm}^2$ (shown in figure 1.2c by the arrow labeled A). The thin solid line represents the voltage component of the stable equilibrium point as the current varies. The thin dashed line represents the unstable equilibrium. At about $I = 95 \mu\text{A}/\text{cm}^2$, the equilibrium becomes unstable. At this point, a new solution appears; it is an unstable limit cycle (thick dashed line). The maximum and minimum values of the voltage are plotted for each current level. This unstable limit cycle exists only for a small range of currents between around 87 and $95 \mu\text{A}/\text{cm}^2$. For currents beyond $95 \mu\text{A}/\text{cm}^2$, there is a unique stable limit cycle represented by the thick solid curve in figure 1.2d. For currents between 87 and $95 \mu\text{A}/\text{cm}^2$, there is both a stable equilibrium and a stable limit cycle.

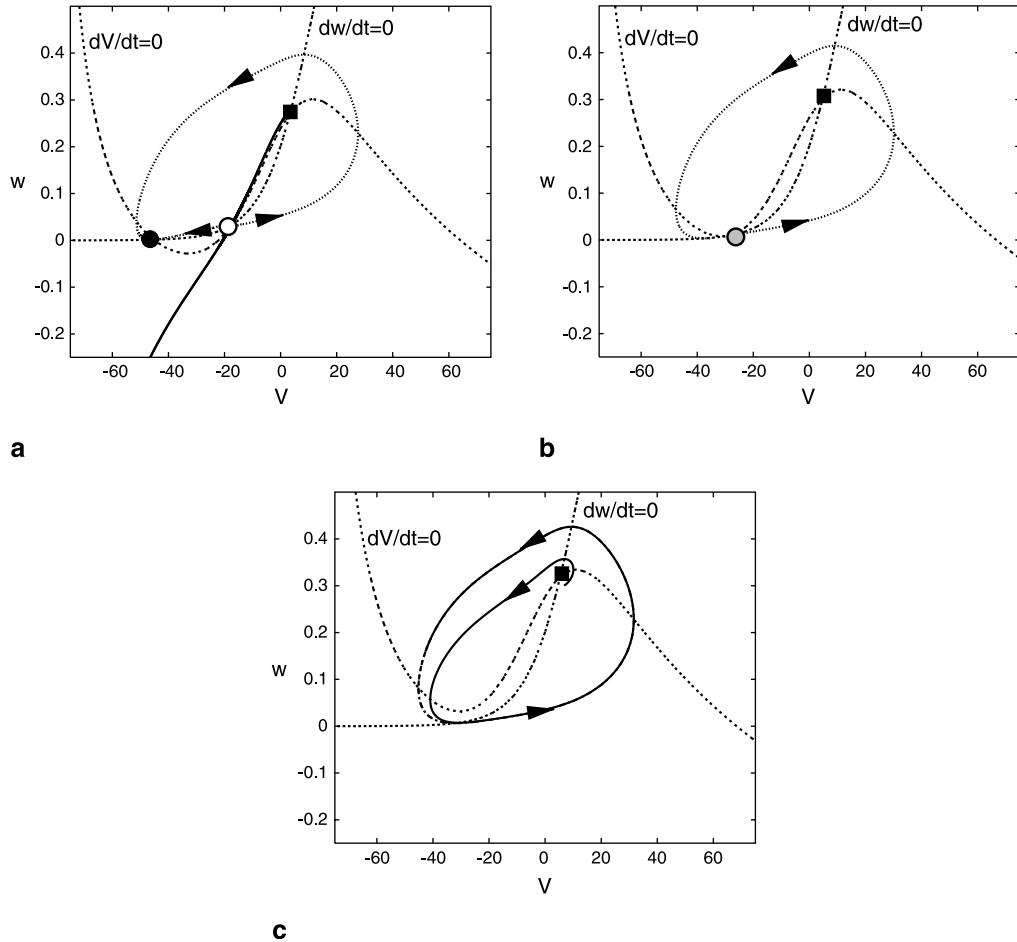


Figure 1.3
Saddle-node excitability.

Amazingly enough, there are only a very few ways in which an equilibrium point can give rise to a limit cycle as a parameter varies. The Hopf bifurcation described here is one such example, which occurs commonly in models of neurons. In fact, the Hopf bifurcation is the mechanism by which the Hodgkin-Huxley squid axon model makes the transition from being at rest to repetitive firing. There is another common mechanism through which neurons go from a resting state to regular firing called the *saddle node on an invariant circle* or simply, SNIC bifurcation. Figure 1.3 illustrates this bifurcation in the model (equations 1.30 and 1.31) with different parameters from figure 1.2. At low values of the current in figure 1.3a there are three equilib-

rium points: a stable node (the black circle), an unstable node (black square), and a saddle point (white circle). Contrast this with figure 1.2, where there is always one equilibrium point. The resting state is the stable node. A diagonal line enters the saddle point and forms a true threshold curve for the model. Any perturbations that move the voltage to the right of this curve from the resting state will result in an action potential. As the current increases, the saddle and the stable node coalesce at a saddle-node bifurcation (figure 1.3b, gray circle). There is now a limit cycle, but it contains an equilibrium point, so that it has an “infinite period” or zero frequency. A small increase in the current lifts the V -nullcline past the w -nullcline (figure 1.3c), allowing the system to fire repetitively. Since lowering the current can bring the nullclines arbitrarily close together, it can take a very long time to get through the narrow gap and thus the frequency of this model can be arbitrarily low. No bistability is possible near the SNIC, in contrast to the Hopf bifurcation. Many models for regularly spiking pyramidal cells have this bifurcation. Further coverage on the SNIC, Hopf, and other bifurcations in neuronal models can be found in Izhikevich (2006) and Rinzel and Ermentrout (1998).

We have only scratched the surface of dynamic possibilities, even for 2-D ODE systems. Our main goal here has been to demonstrate the benefits and insights obtained from a geometric approach and how you can predict qualitative behavior if you develop a little intuition about how the nullclines depend on system parameters. For example, you could easily imagine that changing some parameters could distort the nullclines and lead to multiple intersections (therefore, multiple steady states) and possibilities for multistability and SNIC bifurcations.

Although the phase-plane treatment is completely thorough only for 2-D systems, one can use the concepts to gain insight into the behavior of some higher-order systems; e.g., by looking at projections that show a fast and a slow variable. Also, if one variable z is very slow, then one may consider the phase plane as parameterized by z , and snapshots or animations of the phase plane can be illuminating. The bifurcation phenomena (Hopf, SNIC, saddle node) are general, occurring in higher-dimensional systems, beyond 1-D and 2-D cases.

1.4 Space, Time, and Partial Differential Equations

One issue confronting a modeler is when to use ODEs for a system. We consider two situations where there might be some questions. Suppose that we wish to model the electrical activity of a neuron with dendrites or an axon.

A typical neural model involves many variables, such as the potential, the calcium concentration, and the gating variables. Is it reasonable to assume that the neuron is equipotential? That is, do we expect the potential in, say, the soma, to be the same as that halfway along the apical dendrite? If the neuron is electrically compact, then this

is probably not a bad approximation. However, if channel densities vary wildly or the neuron is spatially extended, then it is probably better to subdivide the neuron into compartments and write differential equations for each of these (see chapter 11, section 1). Reducing a neuron to a single compartment is making the implicit assumption that the neuron is space clamped. Similarly, if we are modeling chemical reactions in a beaker, a common assumption is that the chemical species are well mixed, so that there are no spatial differences. Then it is possible to write a single differential equation for each of the reacting species (see chapter 3). In both the neuron example and the chemical example, we can write a single differential equation for each variable if we believe that the spatial differences are not important. In many modeling studies, this is a good first step. On the other hand, if we think that spatial differences matter, then it is necessary to write equations for the variables at many (possibly infinitely many) spatial locations. Figure 1.2A showed an example of a typical excitable membrane model.

Suppose we want to model a piece of axon to assess how long it takes the action potential to propagate over some small distance. Then, clearly, we should not treat the axon as space clamped or well mixed. Instead, we subdivide the axon into N segments or compartments of length Δx and assume that they are connected to their nearest neighbors via the intracellular resistance. Each compartment obeys equations (1.30)–(1.31). We can thus write:

$$\begin{aligned} c \frac{dV_j}{dt} &= \frac{D}{\Delta x^2} (V_{j+1} - 2V_j + V_{j-1}) + I_0 - I_{\text{ion}}(V_j, w_j) \\ \frac{dw_j}{dt} &= \frac{w_\infty(V_j) - w_j}{\tau_w(V_j)}, \end{aligned} \tag{1.32}$$

where I_0 is the applied current (applied to every compartment) and

$$I_{\text{ion}}(V, w) = g_L(V - E_L) + g_{\text{Ca}}m_\infty(V)(V - E_{\text{Ca}}) + g_Kw(V - E_K) \tag{1.33}$$

represents the membrane ionic currents in the axon. We mention a few interesting points that these equations bring out; for other details, see chapter 10, section 1. First, $D = a/(2 \times R_{\text{axial}})$ where a is the radius of the dendrite and R_{axial} is the specific axial resistivity (typically measured in $\Omega\text{-cm}$, so that D has dimensions of conductance). Second, we have written this equation in terms of an index j corresponding to the j th compartment. Interactions occur with the neighbors, $j \pm 1$. So, it is natural to ask what happens at $j = 1$ and at $j = N$. For example, at $j = 1$, there is a reference to V_0 so that in order to make sense of the equation when $j = 1$, we have to specify V_0 . Similarly, at $j = N$, we need to specify V_{N+1} . The values for these “boundary” points are the boundary conditions for the model. There are many choices for boundary conditions possible.

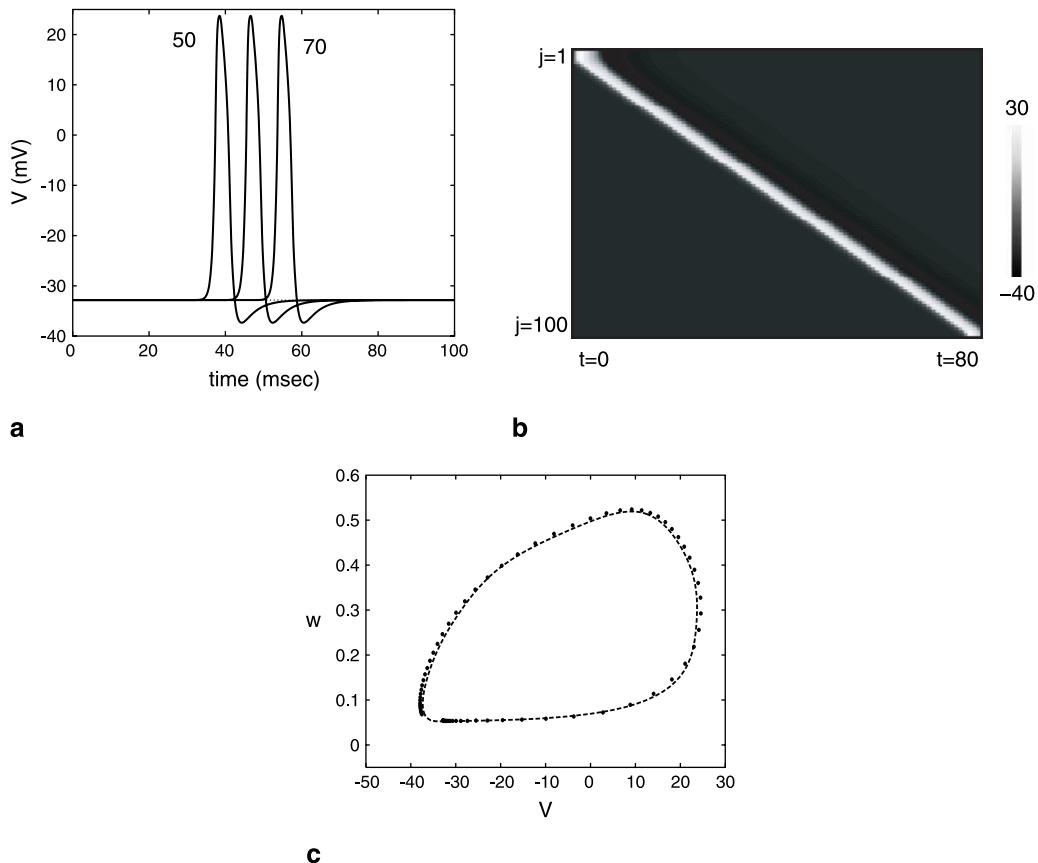


Figure 1.4
Partial differential equations.

Figure 1.4 shows a simulation of an axon broken into 100 compartments, $D/\Delta x^2 = 1$. We have set $V_0 = V_1$ and $V_{101} = V_{100}$ for this simulation, which corresponds to clamping the current flow out the ends to zero. We start all compartments at their resting value except the first three, which start above the threshold. The result is a wave propagating to the right. Unlike the space-clamped equation, activity persists for a much longer time (80 ms compared with 10 ms) and if the cable were longer, the activity would persist for an even longer time. Figure 1.4a shows the potentials in three different compartments, 50, 60, and 70. The time difference between the action potential at $j = 50$ and $j = 60$ is exactly the same as that between $j = 60$ and $j = 70$. This is the hallmark of a traveling wave.

Figure 1.4b shows the full space-time picture. Time increases from left to right and compartment number increases from top to bottom. The potential is shown as a

grayscale and traces out a straight line except near the ends of the cable. If we fix the length of the axon to be l and let the number of compartments, N , go to infinity (that is let $\Delta x = l/N$ go to zero), then equation (1.32) becomes a *partial differential equation (PDE)*:

$$\begin{aligned} c \frac{\partial V(x, t)}{\partial t} &= D \frac{\partial^2 V(x, t)}{\partial x^2} + I_0 - I_{\text{ion}}[V(x, t), w(x, t)] \\ \frac{\partial w(x, t)}{\partial t} &= \frac{w_\infty[V(x, t)] - w(x, t)}{\tau_w[V(x, t)]}. \end{aligned} \quad (1.34)$$

As with an ordinary differential equation, the equations define the rates of change of the voltage and the potassium channels at each spatial location $0 < x < l$. The interaction in space is via the term $\partial^2 V / \partial x^2$, which represents two derivatives with respect to space. As with equation (1.32), we need to define conditions at the ends of the axon, $x = 0$ and $x = l$. The boundary conditions corresponding to the simulation in figure 1.4 can be written as

$$\frac{\partial V(0, t)}{\partial x} = \frac{\partial V(l, t)}{\partial x} = 0. \quad (1.35)$$

Physically, this means that no current can flow out the ends of the cable. In addition to the boundary conditions, initial conditions are also required. That is, $V(x, t_0) = V_0(x)$, $w(x, t_0) = w_0(x)$, where V_0 , w_0 are some functions. As with the ODE, under reasonable assumptions there is a unique solution to this PDE for some positive time.

Looking at figure 1.4a, it would appear that the potential at a point (x_1, t_1) is exactly the same as the potential at a point (x_2, t_2) where $v \equiv (x_2 - x_1)/(t_2 - t_1)$ is constant (at least away from $x = 0$ and $x = L$). That is, the action potential takes the form of a constant-velocity wave moving to the right. The velocity is the constant v , which for the present simulation is roughly 1.25 space units per millisecond (that is it takes 80 ms to travel 100 compartments). This observation allows us to convert the PDE into a special kind of ODE. First, we make the axon infinite in length since we want the wave to travel forever. This is a mathematical conceit that allows us to simplify the equations a great deal. The way to think of this is that when the traveling action potential is in the middle of the axon, the behavior at the ends is essentially irrelevant to the dynamics in the middle. Let $y = x - vt$ be a moving coordinate system; a large and negative y corresponds to regions in space and time where the wave has already passed (points at the left of the line in figure 1.4b) while a large and positive y corresponds to regions that have not yet experienced the activity. Let $V(x, t) = u(y)$ and $w(x, t) = z(y)$ be the potential and the potassium

activation in the moving frame. Then the PDE, equation (1.34), can be written as an ODE:

$$\begin{aligned} -vc \frac{du}{dy} &= D \frac{d^2u}{dy^2} + I_{\text{ion}}(u, z) \\ -v \frac{dz}{dy} &= \frac{w_\infty(u) - z}{\tau_w(u)}. \end{aligned} \tag{1.36}$$

Unlike the ODEs we have encountered before, we do not want to specify initial conditions for this. Instead we want the voltage and the potassium gates to be at rest at the far ends of the axon and we want to specify a particular value of the potential in the middle of the cable, say, $V = 0$. This latter condition simply sets the origin for the wave since it is translation invariant. Thus, we want

$$u(\pm\infty) = V_{\text{rest}}$$

$$z(\pm\infty) = w_{\text{rest}}$$

$$u(0) = 0.$$

As we noted earlier, it is always possible to satisfy the last condition as long as $u(y)$, $z(y)$ is not identical to the resting state. ODEs where conditions are specified at different values of the independent variable are called *boundary-value problems* (BVP) and are considerably harder to solve both analytically and numerically. Equation (1.36) can be written as three first-order equations (as we did in the earlier part of this chapter). In general, three ODEs means we can specify at most three conditions. However, we have four conditions (both u, z must go to rest as $y \rightarrow \pm\infty$). However, there is a free parameter, the velocity, v . Thus we want to know if there is a value of v such that the solutions to equation (1.36) satisfy the four boundary conditions.

Unlike initial-value problems, there is no guarantee that there is a solution to the BVP. Through a method called shooting, it is possible to estimate the value of the velocity $v \approx 1.34$ so that the four conditions at $\pm\infty$ are met. We remark that this value is within 10% of our crude estimate from figure 1.3a. Figure 1.3c shows the projection onto the $V - w$ phase plane of the solution to the compartmental model at $j = 50$. Superimposed on this as a series of dots is the solution to equation (1.36). Thus we see that while shooting requires some work, it provides a good description of the discretized solutions to the full PDE. In their pivotal paper on the squid action potential, Hodgkin and Huxley (1952) also used shooting to estimate the velocity of the action potential since solving the full PDE was beyond the computational capabilities at the time. In conclusion, PDEs are necessary when spatial differences matter; PDEs require both initial and boundary conditions; and certain forms of solutions to PDEs can be reduced to ODEs.

1.5 Noise

Let us turn to chemicals reacting inside a cell (see chapter 3) or channels opening and closing in a small patch of membrane (see chapter 5). Within the cell, some chemical species may be represented by as few as one or two molecules, such as messenger-RNA (mRNA) for some protein. In this case, it does not make any sense to write a differential equation for the concentration of the mRNA. Similarly, in a tiny patch of membrane (e.g., on the head of synaptic spine), there may be only a few dozen channels and so it becomes questionable whether to treat the fraction of open channels as a continuous variable. Rather, we may have to treat each species or channel as a random individual. (The randomness arises because the opening and closing of individual channels is typically a stochastic process.) For a large patch of membrane or in a larger environment, there are many more channels or molecules and thus, if they are acting independently, it makes sense to treat the fraction or concentrations as continuous variables and thus differential equations. However, in low concentration or when there are few channels, the fluctuations [which die off like $1/(M)^{1/2}$, where M is the number of molecules or species] can dominate the behavior and e.g., produce finite-time extinction or oscillations where none occurred for the continuous model. Still, we may think of this as a well-mixed system with only a finite number of channels or molecules.

Returning to our old friend the Morris-Lecar equation, we replace the dynamics for the potassium channel activation in equation (1.31) with a stochastic value determined by the following dynamics: we suppose that there are M independent potassium channels. Typically, channel opening and closing is treated as a Poisson process with a rate that is a function of the voltage. This means that if the rate of opening is α , then the probability that a channel will go from open to closed in the time interval Δt is $\alpha\Delta t$ for small-enough time intervals. Similarly, an open channel will return to a closed state with the probability $\beta\Delta t$. If α or β is large, then we must make the time increment Δt very small. We use parameters exactly as in figure 1.2a, but have only eighty potassium channels that are governed by the random opening and closing of the gates with

$$\begin{aligned}\alpha(V) &= w_\infty(V)/\tau_w(V) \\ \beta(V) &= (1 - w_\infty(V))/\tau_w(V).\end{aligned}\tag{1.37}$$

At any given time there are, say, p open channels so that $w = p/M$ represents the fraction of open channels and is thus used for the w in the voltage equation (1.30). Figure 1.5a shows a typical simulation of the stochastic model. Note that the fluctuations are sufficiently large to cause the neuron to fire spontaneously even though there are no stimuli. Zooming into one of the spikes (figure 1.5b), we can see the dis-

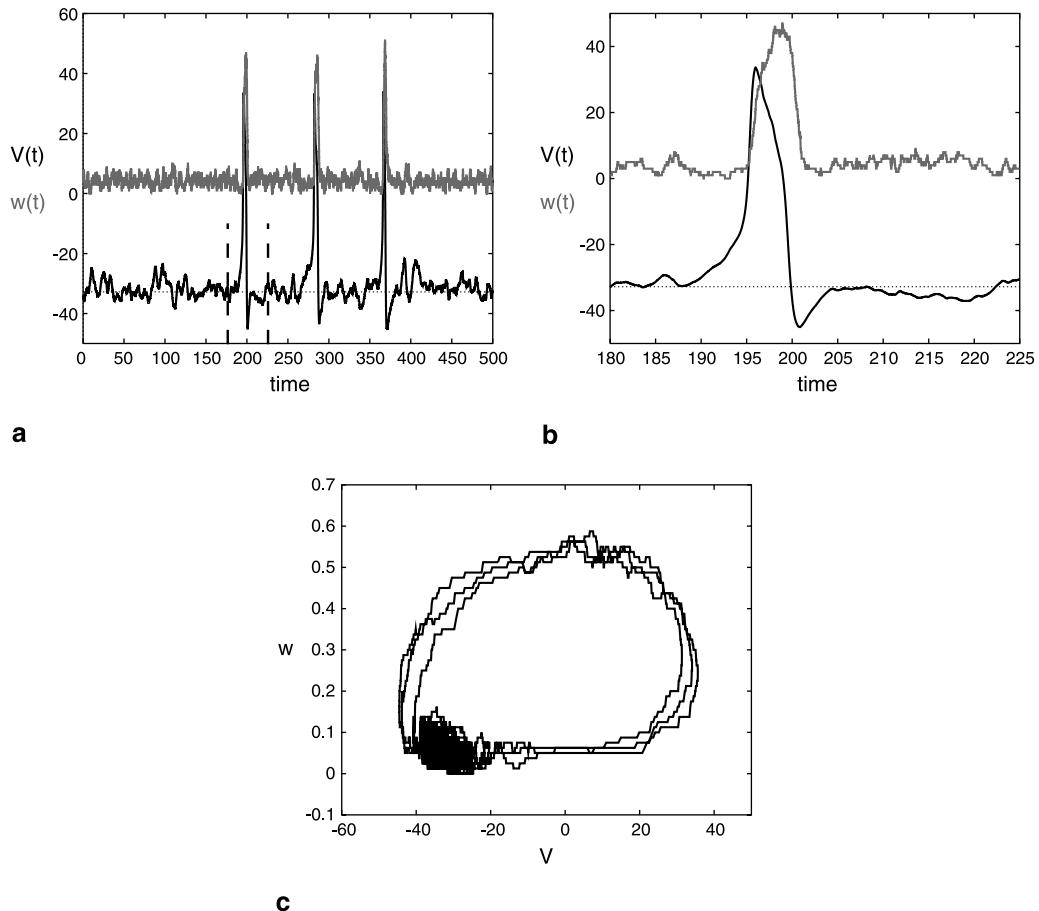


Figure 1.5
Noisy ODEs.

crete increments taken by $w = p/M$. This is even more apparent in the phase-plane plot shown in figure 1.5c; w is constant while V moves according to the deterministic dynamics of equation (1.30). With fewer channels, say twenty, the membrane is much more active and instead of producing spikes at a rate of about 6 Hz, it produces them at a rate of 30 Hz. Thus, in some instances, stochastic fluctuations matter a great deal in the qualitative behavior of the modeling. Here, with deterministic dynamics, the neuron is “dead,” but with small numbers of stochastic channels, it springs to life (similar behavior is seen in the Hodgkin-Huxley model; Strassberg and DeFelice, 1993).

1.6 Numerical Methods

Very few differential equation models for neural behavior admit a closed-form solution in terms of simple functions. Thus most neural models are simulated by solving them numerically on a computer. The numerical solution of ODEs is a mature field and there are many good packages available for solving initial-value problems. Boundary-value problems are more difficult to solve and their solution remains an area of active research. PDEs are even more difficult to solve. For the kinds of PDEs in neuroscience, such as models for dendrites and axons, the easiest way to solve them is to divide them into small isopotential compartments and then solve the resulting large system of ODEs (see chapter 11, section 1). This is precisely what programs like NEURON and GENESIS do to simulate reconstructed neurons (see the software appendix).

ODEs

Here we briefly discuss methods for solving ODEs using a one-dimensional model as an example:

$$u' = G(u, t). \quad (1.38)$$

The easiest way to approximate this is to recall the finite-difference approximation:

$$\frac{\Delta u}{\Delta t} = G(u, t), \quad (1.39)$$

which says that the change in u is approximately equal to

$$\Delta u \approx \Delta t G(u(t), t). \quad (1.40)$$

Let us (for notational convenience) denote Δt by the parameter h , which we will call the step size of the approximation. Suppose we know u at t_0 . Let $t_1 = t_0 + h$ and more generally, $t_n = t_0 + nh$. Then we can approximate the function $u(t)$ at these values, $u_0 = u(t_0), u_1 = u(t_1), \dots, u_n = u(t_n)$, by the iteration:

$$u_{n+1} = u_n + hG(u_n, t_n). \quad (1.41)$$

This iterative scheme, called *Euler's method*, is the easiest to implement and is therefore quite popular although it is one of the most inefficient methods; in order to be accurate, h must be very small. At each step, the approximation gives an error that is roughly of size h^2 , so that at the end of a simulation the error is roughly proportional to h .

An improvement on Euler's method requires making multiple evaluations of the right-hand side, $G(u, t)$, for each step taken. For example, the well-known *Runge-*

Kutta method requires four evaluations at each time step. (The Runge-Kutta method is essentially Simpson's rule for integration; another simple method requiring two right-hand-side evaluations is *Heun's method* and is equivalent to the trapezoidal rule.) The reader might well wonder why one would want to do more work. It turns out that the Runge-Kutta method gives an error that is proportional to h^5 at each step and proportional to h^4 at the end of the simulation. Suppose that you wanted an error of 10^{-4} at the end of a simulation that lasted, say, for 1 s. So for Runge-Kutta, you take $h = 0.1$ and it requires 40 evaluations of the right-hand side. Euler requires $h = 0.0001$ for the same accuracy and thus needs 10,000 evaluations of the right-hand side. So, for a little extra coding work, you get an incredible increase in speed.

Since Runge-Kutta requires four right-hand-side evaluations, it is called a *fourth-order method*. By comparing the results of, say, a fourth-order method and a fifth-order method, it is possible to get an excellent estimate of the error for that time step. If the user prescribes a fixed maximal error, it is then possible to change the time step during the computation to stay within the error. When the equations are rapidly changing (G is large), take small steps, but when G is small, take large steps. Writing so-called adaptive step-size solvers is not as easy as the fixed-step integrators, but they can dramatically increase the speed of solving ODEs. There are many adaptive step-size packages available, so most users need not worry about this.

Accuracy is one important issue in solving ODEs. There is another important issue, *stability*, which roughly describes whether the iteration scheme will converge to a solution near the true solution. Consider

$$x' = -ax, \quad x(0) = 1, \quad (1.42)$$

where $a > 0$ is large. The solution to this is $x = e^{-at}$. The Euler approximation to this is

$$x_{n+1} = x_n - ahx_n = (1 - ah)x_n. \quad (1.43)$$

Suppose that $ah > 1$ so that $1 - ah < 0$. This means that at each step the sign of x_n switches. This is far removed from the true behavior. One way to rectify this is to make h very very small, but this will then take a very long time to simulate. Equations where there is an equivalent of a very large value of a (e.g., some components in a system relax much faster than others) are called stiff; they call for different methods. Suppose that we change the Euler method slightly to

$$x_{n+1} = x_n - ahx_{n+1}. \quad (1.44)$$

Instead of putting x_n on the right-hand side, we put x_{n+1} . Solving this for x_{n+1} yields

$$x_{n+1} = \frac{x_n}{1 + ah}. \quad (1.45)$$

This sequence of iterations stays positive and decays to zero no matter how large a is for any step size h ; this method is unconditionally stable. Applying this to our general first-order equation, we must solve

$$u_{n+1} = u_n + hG(u_{n+1}, t_{n+1}). \quad (1.46)$$

We immediately see a problem: we have to solve a possibly nonlinear equation for u_{n+1} :

$$u_{n+1} - hG(u_{n+1}, t_{n+1}) = u_n. \quad (1.47)$$

Solving nonlinear equations is a major area of research in numerical analysis, so that this method, called backward Euler, is best left to the professional. Typically, if h is not too large, we have a good guess for u_{n+1} , namely, the previous value, u_n , so that one can use Newton's method to solve for u_{n+1} . Newton's method requires that we also know the derivative of the function G with respect to u , but this can be done numerically. In general for N -dimensional systems, we have to solve a large system of nonlinear equations. Methods that put x_{n+1} into the right-hand sides are called implicit or backward methods, in contrast to explicit or forward methods. The latter are much easier to code but can fail miserably with certain classes of ODEs that commonly arise. Gear methods and their descendants, such as LSODE and CVODE, combine adaptative and implicit methods to solve stiff ODEs quickly and accurately. More description of numerical methods for simulating differential equations can be found in Mascagni and Sherman (1998).

Boundary-Value Problems

Boundary-value problems arise in cases when we are looking for steady states of PDEs or, as earlier, when looking for traveling wave solutions to nerve axon equations. As we have noted, unlike initial-value problems, there is no guarantee of the existence of solutions. So before embarking on numerical solutions of BVPs, one should first be sure that there is a solution! There are two general strategies for solving BVPs: shooting and collocation. We illustrate the two ideas using a simple steady-state cable equation:

$$-V(x) + \lambda^2 \frac{d^2 V}{dx^2} = 0 \quad 0 < x < 1, \quad (1.48)$$

with

$$V(0) = 1, \quad \frac{dV}{dx}(1) = 0. \quad (1.49)$$

Here λ is not the symbol for an eigenvalue, but is the cable's space constant: $\lambda = [aR_m/(2R_{\text{axial}})]^{1/2}$; R_m is the specific membrane resistivity (in units of $\Omega\text{-cm}^2$) (see chapter 10, section 1).

The idea of shooting is as follows: This is a second-order ODE so we need to specify an initial condition and the derivative at $x = 0$. Choose $V(0) = 1$ to satisfy the initial condition at the left. Now, using the notation V' for differentiation with respect to x , we choose $V'(0) = a$ as the initial condition for V' , where a is just a guess. Next, integrate the ODE until $x = 1$. If $V'(1) = 0$, then we guessed right. Usually, $V'(1)$ will not satisfy this desired condition at $x = 1$; it will either undershoot or overshoot. The idea is to find a value of a so that $V'(1) > 0$ and a different value so that $V'(1) < 0$ and then choose a between these two values, continuing to iterate until convergence is obtained.

Another common way to solve BVPs is to divide the interval [here $(0, 1)$] into m nodes and assign a value of V to each of these nodes. This will result in a division of the BVP into a large set of algebraic equations. These are solved using an equation solver. XPPAUT (see the software appendix) uses this method to solve BVPs. The advantage that this method has over shooting is that it is not sensitive to the stability of the equation's solution.

Noisy ODEs

A common model for noisy neurons adds normally distributed noise to the right-hand sides of an equation, resulting in a stochastic differential equation (SDE):

$$du = G(u, t)dt + dW(t). \quad (1.50)$$

Here dW is the derivative of Brownian motion, also called a Wiener process. We won't formally say anything more about these fascinating models as they are well beyond the scope of this chapter. Rather, we show how to simulate these equations. Suppose that $dW(t)$ has a zero mean and a variance σ^2 . Then the Euler method for simulating this equation has the form:

$$u_{n+1} = u_n + hG(u_n, t_n) + \sigma\sqrt{h}N(1), \quad (1.51)$$

where $N(1)$ is a normally distributed random number with unit variance and zero mean. The unusual part of this model is that the square root of the step size is used to multiply the noise. The effective variance is then σ^2h , which scales linearly with the step size.

Another source of noise is the random flipping of channels. Consider the simple process of a channel flipping between two states:



where O is open and C is closed. The rates are, respectively, α and β . If the channel is in the closed state, then the probability that it opens in time h is determined by a Poisson process and is approximately αh if h is small. We must choose h small enough so that during the same time the chance that it flipped back is essentially zero. Thus the numerical scheme for this type of noise is to pick a random number between 0 and 1. If this number is less than αh , then flip the channel to the open state; otherwise keep it closed. If there are many channels, then this procedure must be applied to each of them independently. This scheme only approximates the dynamics and gets more and more accurate as h gets smaller. We will call this the *Markov algorithm*.

An exact simulation can be obtained by using the *Gillespie algorithm* (see chapter 3, section 3). Let N be the number of closed channels and let M be the number of open channels. Let $r = \alpha N + \beta M$. r is the rate at which some event will happen; that is, one of the N closed channels opens or one of the M open channel closes. We generate a random time for the next event with this rate:

$$t_{\text{next}} = -\frac{1}{r} \ln R_1, \quad (1.52)$$

where R_1 is a random number between 0 and 1. This tells us when the next event will occur. Next we need to figure out which event. Let R_2 be a random number between 0 and r . If $R_2 < \alpha N$, then the event is a closed-to-open transition; otherwise it is an open-to-closed transition. In the former case, decrease N and increase M , while in the latter case do the opposite.

The Gillespie algorithm is exact, so that it is always better than the Markov algorithm. However, if there are only a few channels or the rates are very slow, then the time between events can be very large. If the stochastic system is coupled to some set of ordinary differential equations, then the solution to the ODE between events needs to be computed and using the step size between events can be very inaccurate. Thus, if the rates are very slow or there are very few channels, it is better to use the Markov algorithm, which has a fixed step size. For additional comparisons of algorithms in this context, see Mino et al. (2002).

PDEs

The easiest way to simulate a PDE as in figure 1.3 is to discretize it to a set of ODEs. Consider the passive cable (with time constant, τ and length constant, λ) of length l (see chapter 10, section 1):

$$\tau \frac{\partial V}{\partial t} = -V(x, t) + \lambda^2 \frac{\partial^2 V}{\partial x^2} \quad (1.53)$$

$$V(0, t) = V_a$$

$$\frac{\partial V(l, t)}{\partial x} = 0.$$

We consider the potential at $N + 1$ discrete locations spaced an amount $k = l/N$ apart, and define $V_j(t) = V(j\Delta x, t)$ where $j = 0, \dots, N$. We approximate the spatial derivative by the difference:

$$\frac{\partial^2 V}{\partial x^2} \approx \frac{V_{j+1} - 2V_j + V_{j-1}}{\Delta x^2} \quad (1.54)$$

and obtain the set of ODEs:

$$\tau V'_j = -V_j + (\lambda^2/\Delta x^2)(V_{j+1} - 2V_j + V_{j-1}). \quad (1.55)$$

At $j = 0$ and $j = N$, we need to take the boundary conditions into account. At $j = 0$, we specified the voltage to be $V(0, t) = V_a$, so that the $j = 1$ equation is

$$\tau V'_1 = -V_1 + (\lambda^2/\Delta x^2)(V_2 - 2V_1 + V_a). \quad (1.56)$$

At $x = l$, corresponding to $j = N$, we have the zero-flux boundary condition, which is equivalent to imposing even symmetry about $x = l$. Even symmetry means $V_{N+1} = V_{N-1}$ and then substituting into the differential equation at $j = N$ we get:

$$\tau V'_N = -V_N + 2(\lambda^2/\Delta x^2)(V_{N-1} - V_N). \quad (1.57)$$

This use of even symmetry represents the zero-flux boundary condition more accurately than the one-sided differences for the boundary conditions that we used in the simulation for figure 1.4. Now, we solve this system using our favorite ODE solver. We remark that if N is large, then the step size Δx is small and the parameter $\lambda^2/\Delta x^2$ can be very large. Thus, stiff ODE solvers are often used for PDEs that arise from the cable equation.

Acknowledgments

Bard Ermentrout is supported in part by the National Institute of Mental Health and the National Science Foundation. John Rinzel is supported in part by National Institute on Deafness and Other Communication Disorders/National Institutes of Health, grant DC008543.

2 Parameter Searching

Pablo Achard, Werner Van Geit, and Gwendal LeMasson

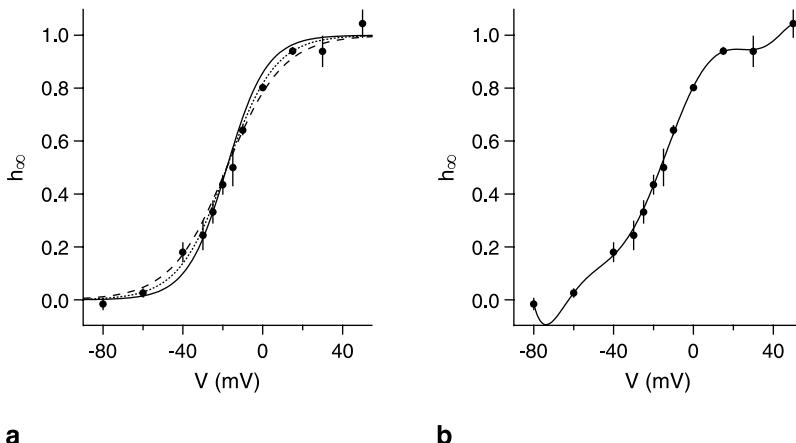
Whatever simulation you want to build, from a single ionic channel to a full cortical column, your model will contain parameters. In the best case, good experimental data will give you precise values for all of them. In the worst, you will have to approximate orders of magnitude and the best you can do are guesses. Most scientists optimize parameters with the familiar trial-and-error method and after several years become real experts in fine tuning. Today processors are quite cheap and you can let computers perform many trials (and make many errors) for you. How to tune parameters in an efficient way is the subject of this chapter. The related issue of uniqueness of parameters and natural variability (Marder and Goaillard, 2006; Achard and De Schutter, 2006) is considered in section 3 of chapter 12.

2.1 Fitness Measures

Before running any optimization algorithm, one must define what is a “good” and what is a “bad” model. During the search process, every algorithm needs to know if a set of parameters—also called a point (in the parameter space), a model, or an individual—is “better” or “worse” than another one. This is done through the calculation, for every model, of a single real number that is usually designated fitness, but also distance, error, or cost. In this chapter, for sake of simplicity we will equate lower values of this number with better fitness and therefore our goal is to minimize fitness values. Fitness functions will be denoted f , while $f(x)$ will be the fitness value for the set of parameters x . Fitness functions will be introduced for two concrete and common examples in the field of neuroscience: activation curves and electrophysiological traces.

Distance between a Boltzmann Function and Channel Activation Measurements

In the Hodgkin-Huxley model (Hodgkin and Huxley, 1952 and chapter 5, section 2) of voltage-dependent ionic channels, the channel activation h_∞ is a sigmoid function (also called a Boltzmann function, B) with respect to the membrane voltage, V :

**Figure 2.1**

Different fits for the activation-voltage relationship. Data (dots) are compared with (a) sigmoids with $k = 10$ (solid line), $k = 12$ (dotted line), $k = 14$ (dashed line), or (b) with a 9-degree polynomial.

$$h_\infty(V) = B(V) = \frac{1}{1 + \exp(-(V - V_{1/2})/k)}, \quad (2.1)$$

where $V_{1/2}$ is the half-activation potential and k is the slope factor. In figure 2.1a, the points represent h_∞ values experimentally measured for different voltages; $V_{1/2}$ is assumed to be known while k is the variable we want to determine. As a first guess, we have plotted sigmoids with k respectively equal to 10, 12, and 14. Which one is the best? Is the best one good enough? Is there a range of acceptable k values? What is the best possible value of k ? We can't answer these questions without defining a fitness function.

In this example, as in many cases, most of the uncertainty in the experimental measure is in one quantity only: V is measured with a much greater accuracy than h_∞ . So, one simple way to measure the distance between our data and a model is to calculate, in the ordinate, the absolute difference (also called *residual*) between the activation values for each data point:

$$f_1 = \sum_{i=1}^N |h_\infty(V_i) - B(V_i, k)|, \quad (2.2)$$

where the N data points h_∞ are measured at voltages V_i , and $B(V_i, k)$ designates the value of the sigmoid at the point V_i for a given k . Since the derivative of f_1 is not continuous, we often prefer a variant of it, called the *least-square error*, lse (which has additional advantages according to the Gauss-Markov theorem), defined by

$$\text{lse} = \sum_{i=1}^N [h_\infty(V_i) - B(V_i, k)]^2. \quad (2.3)$$

However, these distance definitions lose a part of the information we collected experimentally, namely, the amount of uncertainty (or error) attached to each measurement. Indeed, f_1 and f_2 give the same importance to points whether or not they are accurately measured (see the differences between the lengths of the error bars in figure 2.1a). Weighting each residual with the inverse of the error can easily rectify this. If errors are measured as the variance over repeated measurements, then this gives the χ^2 (*chi-square*) value:

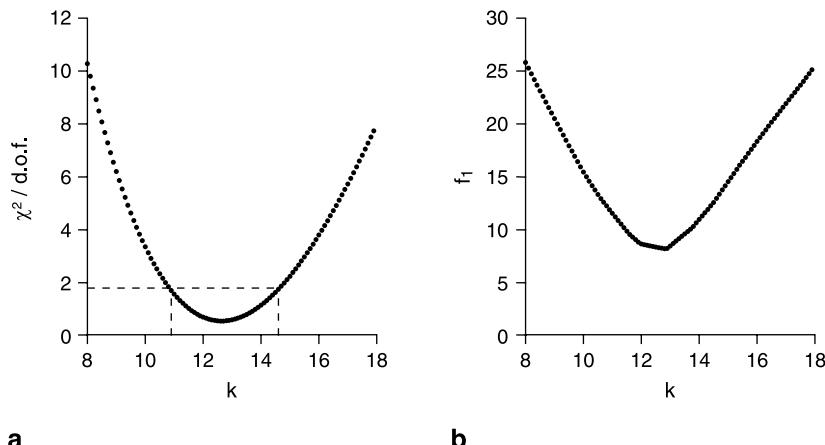
$$\chi^2 = \sum_{i=1}^N \left(\frac{h_\infty(V_i) - B(V_i, k)}{\text{var}_i} \right)^2 \quad (2.4)$$

Using this last definition, the curves of figure 2.1a have a fitness equal to 36.9, 7.7, and 12.7 for $k = 10, 12$, and 14 , respectively. We can therefore answer the first question: the best curve of the three is the dotted one ($k = 12$). Whether this curve is good enough is not within the focus of this chapter, but the interested reader can learn more about the chi-square test in section 2.7.

We also wanted to know if there is a range of k values giving good sigmoids. Our fitness function (χ^2 or $\chi^2/\text{d.o.f}$) can tell us if one sigmoid (a model) is better than another one, but the limit between “good” and “bad” models is always a matter of choice. Let’s specify that a curve will be good if its $\chi^2/\text{d.o.f} < 1.8$, which corresponds here to a 95% confidence limit (see <http://faculty.vassar.edu/lowry/tabs.html#csq>). This definition is arbitrary in the sense that upper limits of 1.5 or 2 would have been good criteria too, but a value of 10 is without doubt out of bounds. We have calculated the $\chi^2/\text{d.o.f}$ of 100 curves with a k value between 8 and 18. The results are shown in figure 2.2a. With this figure, we can define a range of good k values: $10.9 < k < 14.6$. We can also find the best possible value: $k_{\text{best}} = 12.6$. Note that we would have found a similar value for k_{best} if we had used f_1 as fitness function (figure 2.2b). However, the $f_1(k)$ curve is not continuously derivable and is not as smooth. Before describing efficient algorithms to find k_{best} , let us introduce fitness functions for a more complicated but common situation.

Distance between Electrophysiological Traces

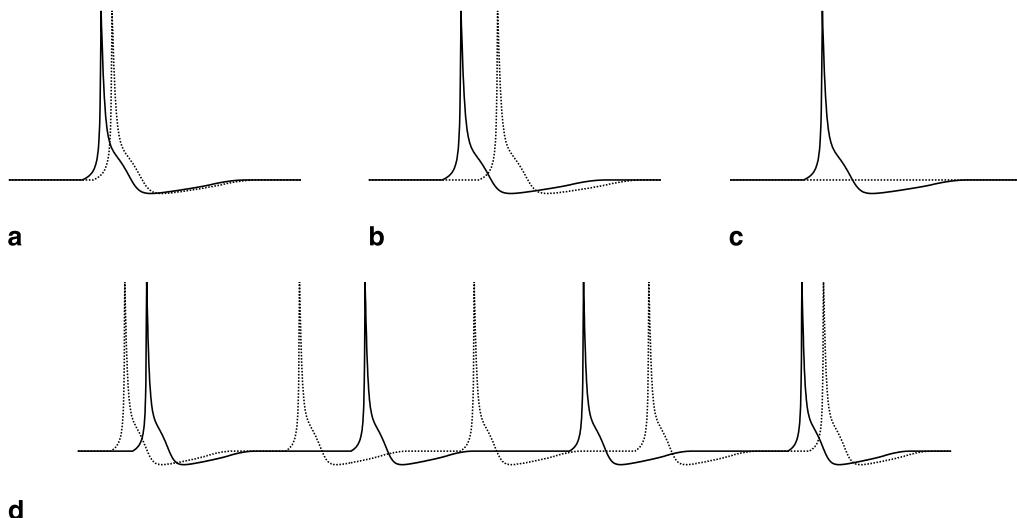
A large amount of data collected in neuroscience laboratories consist of electrophysiological recordings. Consequently, many neuronal models try to reproduce as precisely as possible the electrophysiological traces generated by a given input (however, see Druckmann et al., 2007 and Günay et al., 2008 for an alternative approach).

**Figure 2.2**

Fitness values of sigmoids with respect to their k values. For different values of k , the distance between the sigmoid and the data from figure 2.1 is calculated (a) as a $\chi^2/\text{d.o.f}$ or (b) as an absolute difference. The dashed lines in (a) delimitate the range of good k values.

Comparing data and model traces can be straightforward with a fitness function similar to equations (2.2) or (2.3). Both data and model are time sampled at a certain frequency. If the frequency is the same, the voltage difference is easy to measure at every point in time. If not, it is not much harder to take into account only simultaneous points or to extrapolate additional points on the voltage trace with the lowest sampling. However, the inherent nonlinearity of the neuron responses makes such a fitness function quite unreliable. For example, in figure 2.3, where the data are shown in solid lines and the model in dotted lines, most of us would prefer model a to model b and model b to model c because model b has a larger time shift than model a and model c does not spike at all. However, the values for a fitness function like equation (2.1) are 2.67, 4.76, and 2.92, respectively for a, b, and c. The point-by-point difference is higher for b than for c: the fitness function does not consider the existence of a model spike similar to the data spike at a different time better than no spike at all. Obviously this is something that we would like to avoid.

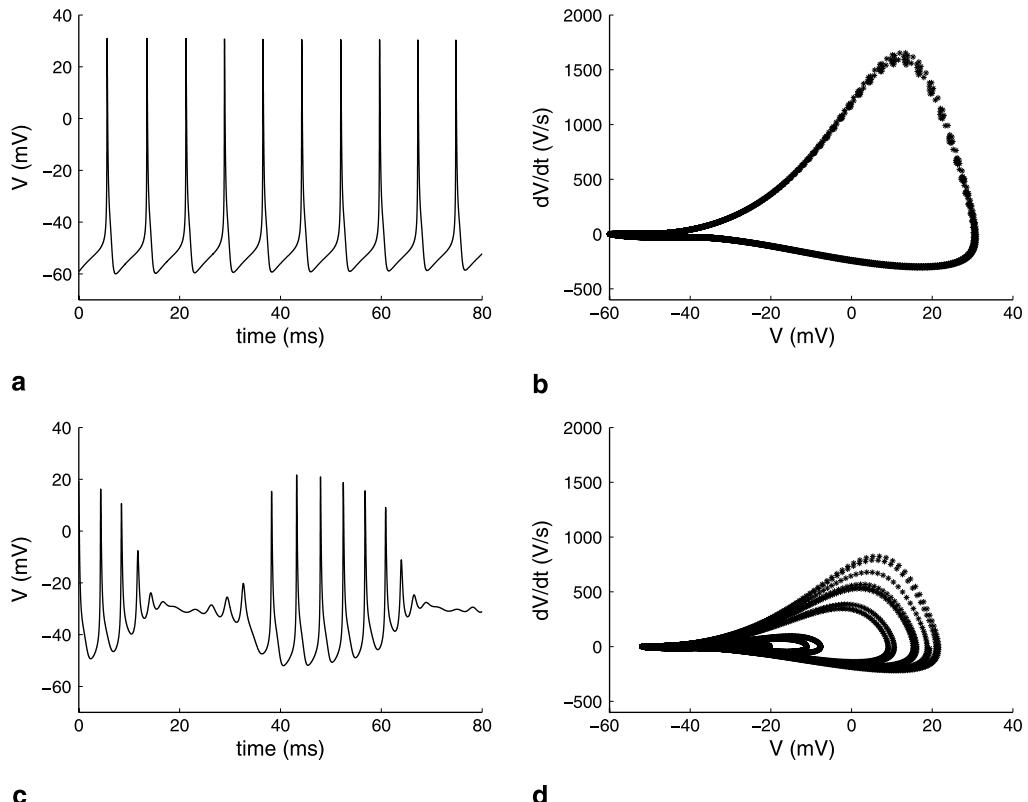
A first possible solution is to shift the model trace so that both peaks coincide, or to compute the point-by-point difference of the voltage trace in a normalized interspike interval using the fiducial-point distance (see Ambros-Ingerson et al., 2008). This method looks simple but rapidly becomes complicated when you deal with traces with more than one spike. Which spikes should be compared (see figure 2.3d)? What happens when the spiking frequencies are different? How does one treat bursts with an irregular number of spikes? And so on.

**Figure 2.3**

Comparing spikes. The data (solid line) are compared with models (dotted lines) (a, b) with the same behavior but with different time shifts, or (c) with a silent model. In (d), pairing data and model spikes is easy for the first spike but is difficult for the following ones.

A good alternative is to build a *feature-based fitness function* that focuses on quantitative measurements of features that are important for physiological function. In many cases these are global activity (e.g., silent, spiking, bursting, synchronicity), resting potentials, spiking frequencies, spike heights, spike widths, after-hyperpolarization potentials, bursting frequencies, number of spikes per burst, and so on (Druckmann et al., 2007; Günay et al., 2008). To do this, you have to measure your favorite properties both in the data and in the model traces. Then you calculate the fitness by using an absolute difference, a least square or even a χ^2 -like function if you measure errors.

One drawback of this method is that it requires criteria to consistently define spikes, bursts, “regularly spiking” neurons, or “synchronous neuronal populations.” This might seem straightforward when you look at your data, but some parameter combinations will certainly drive your model into indecisive states: the spikes will become too small to be real ones but too high to be ignored; the activity between bursts will fade the distinction between “tonically spiking” and “bursting”, etc. Worse, you might want to reproduce not only the simple spiking activity of a cell or a network but also its subthreshold behavior—the spikelets of the bursts or the transmission of postsynaptic potentials. With such complex activities, designing a fitness function comes at the cost of an increasing level of complexity in the definition of what has to be measured.

**Figure 2.4**

Electrophysiological traces and phase-plane trajectories. Simple spiking activity (a) leads to a sharp trajectory in the phase plane (b) while bursts (c) populate a wider region of the phase plane (d).

A simpler solution has been proposed by LeMasson (LeMasson and Maex, 2001) with a fitness function based on the *phase-plane trajectory*. The idea is to represent the neuron's activity, not by its membrane voltage V as a function of the time t , as in figure 2.4a, but by its time derivative $dV(t)/dt$ as a function of $V(t)$, as in figure 2.4b. The derivative $dV(t)/dt$ is simply the slope of the voltage trace $V(t)$ and can easily be approximated by

$$\frac{dV(t)}{dt} \approx \frac{\Delta V(t)}{\Delta t} = \frac{V(t+1) - V(t-1)}{2\Delta T}, \quad (2.5)$$

where ΔT stands for the time interval between two voltage measurements. A series of spikes will repeatedly populate the phase plane to draw a fine trajectory similar to that shown in figure 2.4b. A series of bursts as in figure 2.4c will populate a larger

surface of the phase plane, shown in figure 2.4d. Note that the method is exactly the same for both cases and would remain the same for any kind of behavior, without any need for defining what distinguishes a spike, a burst, etc. Note also that a time shift (as in figure 2.3b) will not affect the phase-plane trajectories.

To derive a fitness function from this phase plane, you define a matrix (or two-dimensional histogram) covering it, count how many hits are within each cell, and calculate the cell-by-cell difference between data and model; this is called the *phase-plane trajectory density method*. These are different possibilities for defining the matrix. Depending on the amount of data you have, you can decide to increase or reduce the number of cells or you may want to have cells of unequal sizes to give more importance to a particular region of interest.

In the original definition (LeMasson and Maex, 2001), the difference between data and model was defined as a least square (equation 2.3). However, the power 2 in this definition increases the impact of the most-populated matrix cells, which are those around the resting potential (Van Geit et al., 2007). Instead, an absolute difference (equation 2.1) or a fitness based on square roots:

$$f_2 = \sum_{i=1}^N \sqrt{|\text{data}_i - \text{model}_i|} \quad (2.6)$$

will give more importance to points of the phase space that are populated during spiking activity (but also to noise when the data are noisy). We have compared equations (2.3) and (2.6) for different test cases, and the fitness landscape is usually much smoother, hence easier to solve, with the latter. This is exemplified in figure 2.5.

The disadvantage of the phase-plane trajectory density method is that it loses all temporal information. If you deal with a neuron that first spikes and then bursts, a model that first bursts and then spikes will match perfectly. Dividing the total time into different periods, comparing only identical periods in the phase plane, and then summing the obtained fitnesses can solve this problem (Van Geit et al., 2007). Be aware that using very short periods will also restore sensitivity to time shifts.

This method works efficiently for complex neuronal behavior (Achard and De Schutter, 2006) and has been implemented by us in a software package, called Neurofitter, that includes several optimization algorithms (Van Geit et al., 2007). Neurofitter works with any type of model; it can be used on parallel computers and is available for free (<http://neurofitter.sourceforge.net>).

2.2 Local Parameter Search Methods

Once you have chosen a metric to determine how good a model is, you must find the best possible model in a minimum amount of time. Several optimization algorithms

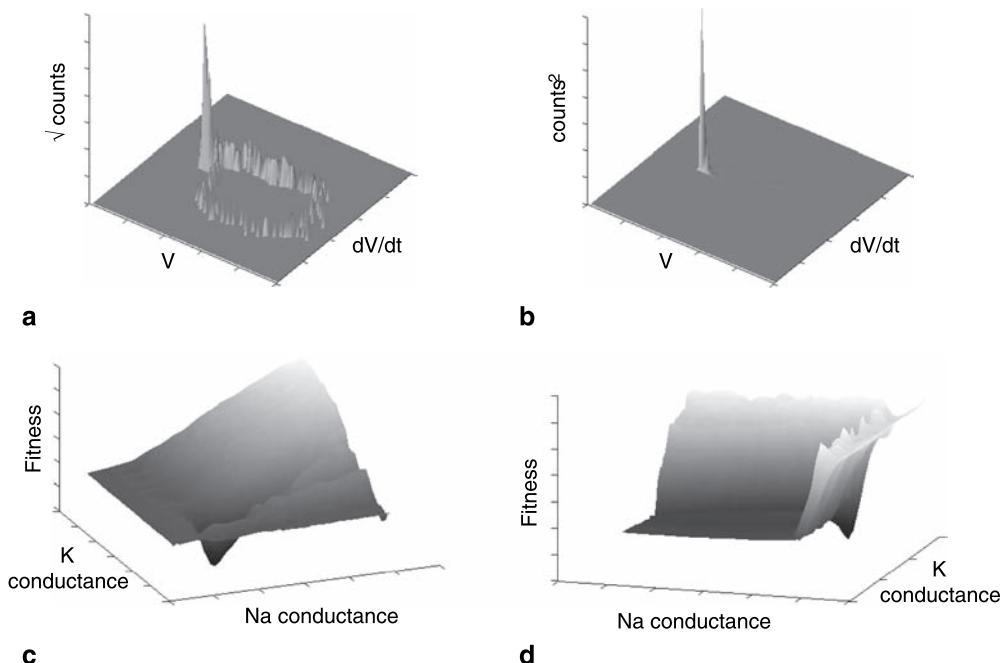


Figure 2.5

Fitness definition and landscape smoothness. The difference between data and model phase planes emphasizes different regions depending on whether the fitness is defined as in equation 2.6 (a) or as in equation 2.3 (b). The fitness landscape is smoother and easier to search in the first case (c) than in the second (d).

exist to perform this task. They can be classified as local or global, deterministic or stochastic, constrained or unconstrained, discrete or continuous, single or multiobjective, static or dynamic, linear or nonlinear, etc. In the following paragraphs, we will restrict ourselves to the most traditional algorithms for current problems and discuss others in section 2.5.

Let us describe first local algorithms that search optima in the vicinity of one or several starting point(s). If the fitness function is convex (as in figure 2.2a, for example), a local search algorithm will have no problem finding the best solution. These algorithms have decades or even centuries of development behind them. They have been implemented in many software packages.

Gradient Methods

The main category of local search methods consists of gradient algorithms. These base their search on the fact that the fitness function f has a null gradient ∇f (or a null derivative f' in one-dimensional problems) when it reaches a local minimum. Among them, we will focus on steepest-descent and (quasi-)Newton methods.

The *steepest-descent method* starts from one point (a) in the parameter space and uses the gradient of the fitness function to find the direction of the next point to evaluate: $b = a - \gamma \nabla f(a)$, where γ rules the length of the move and can vary at each step. The algorithm repeats the operation until $\nabla f = 0$. The biggest difficulty of the method is to find an appropriate value for γ . If it is too small, the algorithm takes a huge amount of time to reach the minimum, whereas if it is too large, the point b may pass over the minimum.

Instead of doing an arbitrary step in the right direction, the *Newton method* searches for $\nabla f = 0$ by repeatedly searching the zero of the tangent of ∇f . It therefore requires the second derivative of the fitness function f'' or, for more than one dimension, its Hessian matrix Hf . The next point to evaluate is then given by the equation $b = a - \gamma [Hf(a)]^{-1} \nabla f(a)$ where γ now plays a minor role and can be fixed. This method is quite similar to the steepest-descent method but uses a quadratic interpolation of f instead of a linear one. It converges much faster but requires the calculation of the Hessian matrix, which can be difficult. To solve this problem, several *quasi-Newton methods* exist that efficiently approximate the Hessian, of which the DFP (for Davidon-Fletcher-Powell; Fletcher and Powell, 1963) and the BFGS (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) are two of the most common methods. In these methods, the Hessian matrix is not computed at every time step but built up step after step. Details of this mechanism are beyond the scope of this book, but interested readers can find more information in Frandsen et al. (2004). Quasi-Newton methods are much more efficient than the steepest-descent method and became popular for simple optimization problems.

Note that for quadratic problems, that is to say, when the fitness function is a second-degree polynomial, there are other gradient methods, such as the Levenberg-Marquardt method (Levenberg, 1944; Marquardt, 1963) or the conjugate gradient algorithm. Since least-square curve-fitting problems belong to this category, these algorithms are also common in data analysis software.

The Downhill Simplex Method

Every time that f is not an analytical function, for example if we have to run a model with a certain set of parameters to measure f , we have no access to the derivative of f , or its computation is time consuming. Nelder and Mead (1965) proposed the downhill simplex method to tackle this problem.

In geometry, a *simplex* is a shape formed by $N + 1$ points, where N is the number of dimensions of the space considered. In two dimensions it is simply a triangle, in three dimensions a tetrahedron, etc. The algorithm starts with a random simplex in the N -dimension parameter space, ranks all the points with respect to their fitness value [$f(x_1) < f(x_2) < \dots < f(x_N) < f(x_{N+1})$], and then iteratively replaces the worst point (x_{N+1}) with a new, better one. To find this new point x' , it first tries the

mirror image of x_{N+1} through the gravity center formed by the N other points [$x_c = (1/N) \sum_{i=1}^N x_i$], which means $x' = 2x_c - x_{N-1}$. If $f(x_1) < f(x') < f(x_N)$, then x' replaces x_{N+1} and the algorithm continues with this new simplex. Instead, if the new point x' is better than x_1 , then a move in the right direction has been made and further point(s) in this direction can be tested: $x'' = \alpha x_c + (1 - \alpha)x_{N-1}$, usually with $\alpha = 3$. Oppositely, if $f(x') > f(x_N)$, then x' will remain the worst point at the next iteration, so a point in between must be tested with the same formula and $\alpha = 1/2$ if $f(x') < f(x_{N+1})$ or $\alpha = 3/2$ otherwise.

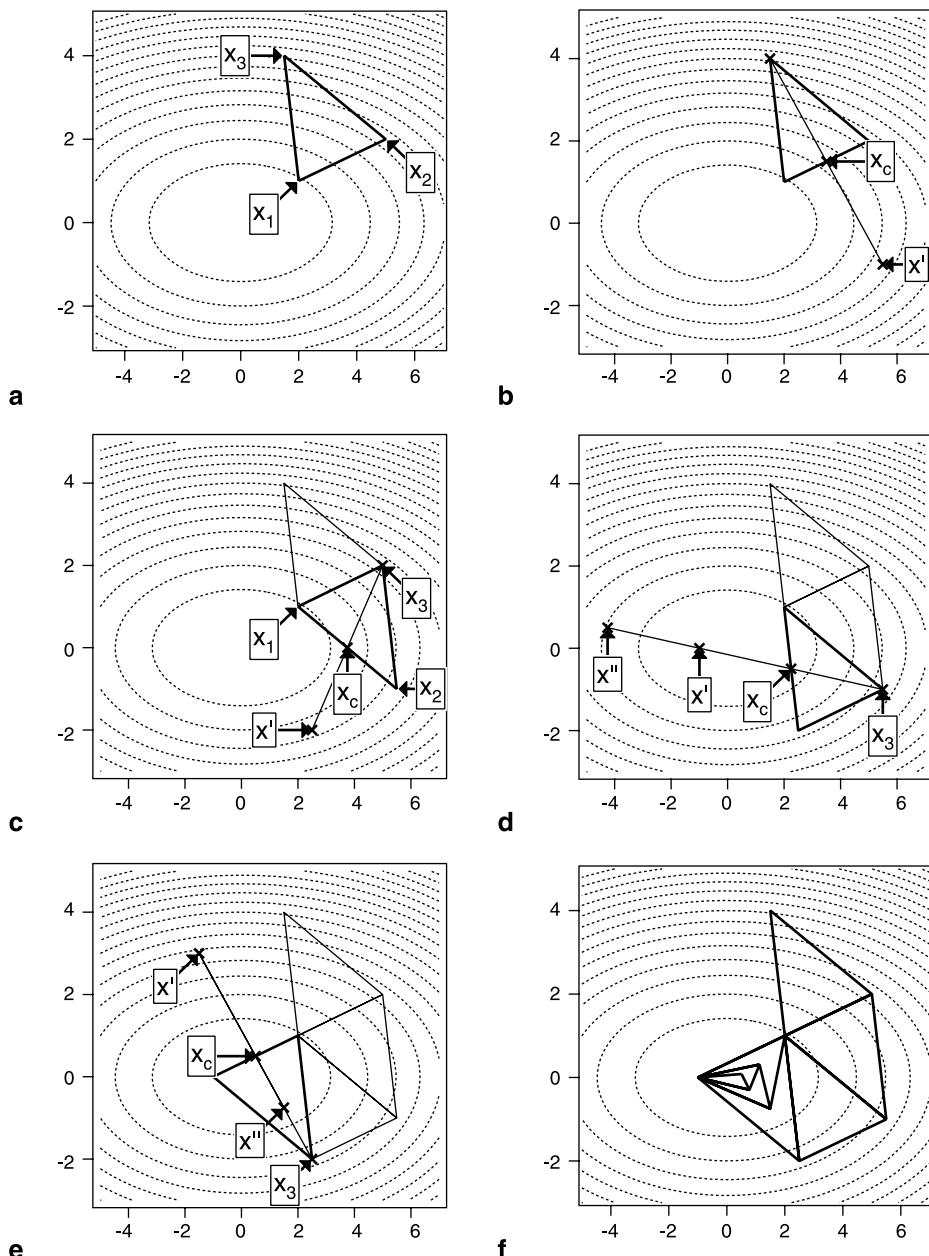
As a two-dimensional example, figure 2.6a shows a fitness landscape with a random simplex (triangle) on top of it. The three corners of the simplex are ordered as a function of their fitness. In figure 2.6b are shown x_c , the center of x_1 and x_2 as well as x' , the mirror of x_3 through x_c . Since $f(x_1) < f(x') < f(x_2)$, x' replaces x_3 and a new simplex is made and drawn on figure 2.6c. The corners are therefore reordered and new x_c and x' points are calculated. The operation is again repeated in figure 2.6d but this time $f(x') < f(x_1)$, so a further point x'' has to be tested. Since $f(x') < f(x'')$, we do not try further and x_3 is replaced by x' . With this new simplex, drawn in figure 2.6e, the opposite situation occurs: $f(x') > f(x_3)$, which drives to a new test point x'' in the middle between x_3 and x_c . Figure 2.6f shows the successive simplexes found by the algorithm that nicely converge toward the minimum, located at $(0, 0)$.

2.3 Deterministic Global Parameter Search Methods

Global Searches: The Exploration–Exploitation Balance

The methods we have seen so far are called local because they search for a minimum of the fitness function in the vicinity of the initial point(s). This is very efficient when the fitness function is smooth. However, in many cases, as in figure 2.5d, the fitness function has multiple points, called local minima, for which the gradient is null. Gradient methods, but also simplex algorithms, are likely to stop at the first point of this kind that they meet. However, you are not interested in finding a local minimum but the global one, i.e., the point for which f is the smallest possible. Global search methods have been developed for this purpose and some of them are described in the following sections.

Because the parameter space is infinite and the global minimum can be located anywhere, the convergence of these algorithms to the global minimum is guaranteed only with infinite time. So, the art of global search algorithms is to find the better possible solution in a reasonable amount of time. Depending on the problem you want to solve, on the time you have, on the computing resources you can recruit, etc., you will favor efficiency (moving as fast as possible to one reasonably good solution) or effectiveness (moving as close as possible to the best solution).

**Figure 2.6**

Successive steps of the simplex algorithm on a two-dimensional problem. Dotted lines represent isofitness curves.

Any algorithm has to find a good balance between exploration of the parameter space and exploitation of the solutions already found. The latter increases the speed of convergence by searching in regions known to contain models with reasonable-to-good fitness, while the former avoids being trapped in local minima by searching in new regions. In a sense, the local search methods we have seen so far are methods for which the exploration–exploitation balance bends to the exploitation side only.

This section describes deterministic global methods and the next one is devoted to stochastic ones. In both, we will keep a careful eye on the exploration–exploitation balance.

The Brute Force Method

The simplest way to explore a parameter space is to systematically scan it. This is done with a grid of M equally distant points in each of the N dimensions of the problem. Since this method leads to M^N fitness evaluations, it is limited to a small value of N , both for speed and memory space reasons (Prinz et al., 2003).

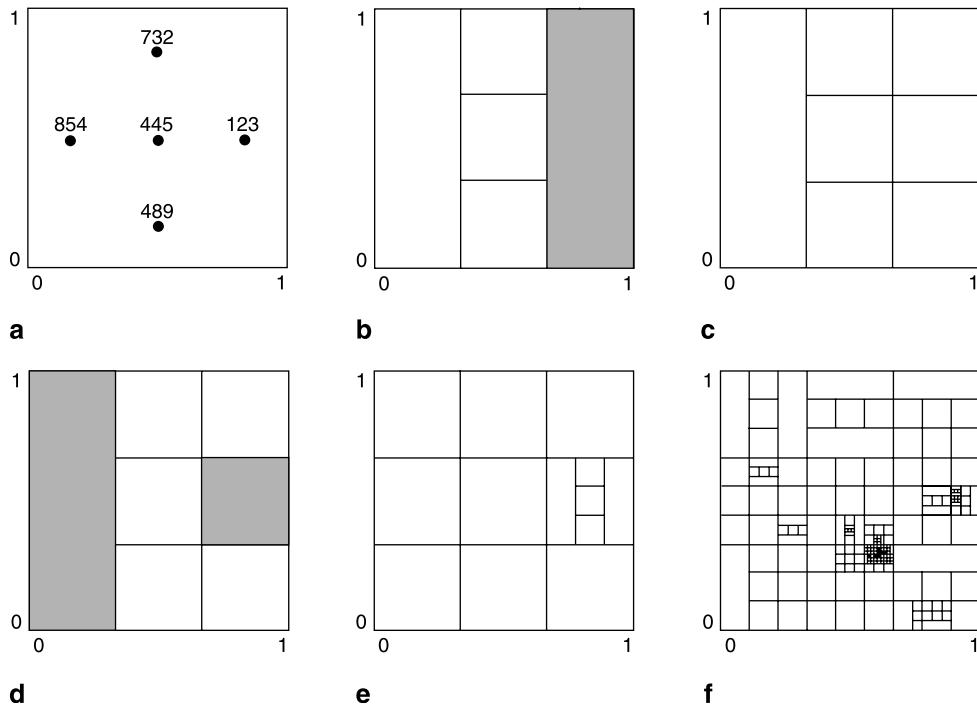
If your fitness landscape is not too complicated, i.e., if your models do not undergo several behavioral changes between two adjacent points of a grid, a mapping of the parameter space might be useful for a better understanding of your model. First, it allows estimating the range of behaviors of the model you are testing: Can your neuron burst? Can some molecule completely disappear? Can your network propagate waves? Second, you can characterize conditions for a certain behavior by revealing continuous or discontinuous regions of the space that lead to the same activity (Prinz et al., 2004a; A. L. Taylor et al., 2006, and chapter 12, section 3). Last but not least, this can give you insights on how parameter change relates to function.

For parameter tuning, this method can only be advised as a first step since the balance is now bending completely on the exploration side. For example, it can be used to reduce the parameter space size for further searches or to give starting points for a local search algorithm.

The DIRECT Algorithm

It is often believed that global search algorithms are always stochastic ones. Let us disprove this assumption with the DIRECT algorithm (Perttunen et al., 1993). For the sake of clarity, the search space that we will examine will be two-dimensional and the parameters (x_1 and x_2) are $0 \leq x_{1,2} \leq 1$. Readers interested in solving higher-dimensional problems with this method can read Finkel (2003).

The algorithm starts by evaluating the fitness of the central point $f(c_1)$ where $c_1 = (1/2, 1/2)$. For each dimension, it then evaluates two more points, so four points in total: $c_2 = (1/2 - 1/3, 1/2)$, $c_3 = (1/2 + 1/3, 1/2)$, $c_4 = (1/2, 1/2 - 1/3)$ and $c_5 = (1/2, 1/2 + 1/3)$, as in figure 2.7a. It then divides the space into five REC-

**Figure 2.7**

Several steps of the DIRECT algorithm. In (a), the algorithm calculates fitnesses of the first five points (fitness values shown). In (b), among the five initial rectangles, one is found as potentially optimal (gray area) and is divided into three squares in (c). In (d), two potentially optimal rectangles are found (gray areas) and divided into three and five in (e). The rectangles obtained after many fitness evaluations are shown in (f).

Tangles (hence its name) with the rule that the point with the lowest fitness must be in the largest rectangle and that the points already evaluated are the centers of the new rectangles (figure 2.7b).

The next step is to find potentially optimal rectangles and simultaneously divide these into smaller rectangles. If the sides of the potentially optimal rectangle are not equal, it is divided into three squares (figure 2.7c). If the potentially optimal rectangle is already a square, it is divided into five rectangles as previously. The procedure is repeated over and over until a stopping criterion is met.

A rectangle is defined as potentially optimal if the fitness of its center is better than the fitness of the centers of all rectangles of the same or larger size. Several iterations of the algorithm are illustrated in figure 2.7.

The exploitation part of the algorithm is provided by the division of the potentially optimal rectangles, while the exploration part is maintained by the rule that fitnesses

are only compared among rectangles of equal or greater sizes, which forces the division of the largest rectangles and avoids underexploration of a priori bad regions.

2.4 Stochastic Global Parameter Search Methods

Stochastic search algorithms use nondeterministic methods to explore the parameter space. The nondeterministic behavior derives from the use of random numbers and therefore the state of the algorithm at a certain time is not completely determined by its previous state. In practice, these numbers are pseudorandom numbers generated by the computer (see section 2.7). This has the advantage that it is always possible to rerun the algorithm using the same seed for the random number generator, to obtain exactly the same results.

Most of the global stochastic search methods share the same core structure. First, one or several initial points x_0 are chosen in the solution space. Then a loop is started with an incrementing variable k , and at every step a number of random points y_k are drawn from a probability distribution Φ_k . This distribution can (but doesn't have to) depend on the previous points x_k . Next a replacement strategy selects which points, among the x_k and y_k , will become the new starting points for the next generation, x_{k+1} .

In its simplest form, new points y_k are chosen uniformly from the solution space. This is called a *blind random search*. Although this method has the advantage of being easy to implement, it does not take into account any information gathered previously from the solution space. In other words, the exploitation is null, so the algorithm is not very effective.

A more efficient version is called a localized random search. Here the new points are calculated as $y_k = x_k + r_k$, where r_k is an independent random vector. Since the random sampling happens around the current best-found solution, it increases the chance of finding better solutions. The risk is that the balance bends too much toward the exploitation side and the algorithm gets stuck in local optima.

When the r_k depends on the values of x_k , i.e., on the best solutions found in the past, the algorithm is called an *adaptive random search*. All the algorithms that follow in the rest of this section are variants of this basic algorithm.

Simulated Annealing

The simulated annealing method is reminiscent of how the structure of a material is enhanced in metallurgy. The process called annealing attempts to create metals with a crystal structure that contains as few defects as possible. The metal is first heated to its melting point so that all the atoms have a large amount of freedom, and then is cooled down very slowly to create crystals. If this cooling process is carried out too

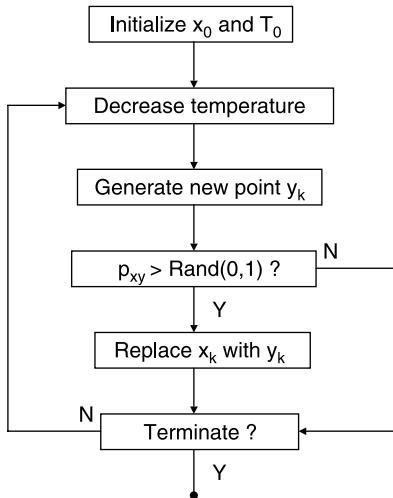


Figure 2.8
The simulated annealing algorithm.

quickly, some parts of the metal may crystallize with different directions, and irregularities in the structure will occur. This kind of process has a lot of similarities with global optimization techniques in that a “cooling” to exploitation follows an initial period of exploration.

The algorithm (Kirkpatrick et al., 1983) consists of the loop illustrated in figure 2.8. It starts with a random point x_0 , and T_0 , the initial temperature of the system. Then, in every loop, the location of a neighboring point y_k is calculated. There are several mechanisms to do this, but the simplest and most-used one is a small random change to each of the parameters. At each step, the probability of replacing the point x_k by its neighbor y_k is given by a transition probability function. One often-used function is based on the Boltzmann-Gibbs distribution:

$$p_{xy} = \begin{cases} 1 & \text{if } f(y_k) < f(x_k) \\ e^{-(f(y_k) - f(x_k))/cT} & \text{otherwise} \end{cases} \quad (2.7)$$

with c a positive constant, T the temperature, and f the fitness function. Consequently the new point y_k will be the starting point of the next step if its fitness is lower than that of x_k . However, it has a non-null probability of replacing x_k even if its fitness is higher. This rule ensures that the algorithm is sometimes able to move uphill to escape from local minima.

The temperature defines how strong the exploration is around the last point. Just as in metallurgic annealing, the temperature is decreased while the algorithm runs

and therefore the balance slowly goes from exploration to exploitation. There is no standard method for doing this. Typical approaches are linear cooling, where the new temperature is calculated as $T_{k+1} = T_k - \delta$ with δ a constant, or exponential cooling, where $T_{k+1} = \rho T_k$ and ρ is a constant. More sophisticated versions, called adaptive simulated annealing, optimize the parameters of the cooling schedule and the way neighbors are created during the algorithm run.

Simulated annealing is relatively easy to implement but has the disadvantage that one has to wait until the end of the cooling before extracting any result. Indeed, the cooling procedure becomes a critical factor that needs to be adapted to the problem you want to solve. If the system is cooled too fast, you will be definitively be trapped in a local minima; if it is too slow, you will waste a lot of time.

Evolutionary Algorithms

Another class of methods, inspired by nature, is based on the principle of natural selection. In this case the “fitness” value of a model is taken literally. The method generates a population of possible models called individuals that evolves over time. Mutating individuals and combining them to create offspring builds consecutive populations. During the evolution, the chance that a model will survive depends on its fitness value (“survival of the fittest”). This way, generation after generation, the probability that better models will be selected increases, just as Darwinian evolution shaped our biological world.

The main algorithm is shown in figure 2.9 and consists of a number of operators represented by rectangles that act upon the population to create new generations. The parent selection operator will select in every generation the individuals that will breed and generate offspring. Applying the recombination operator to parent models

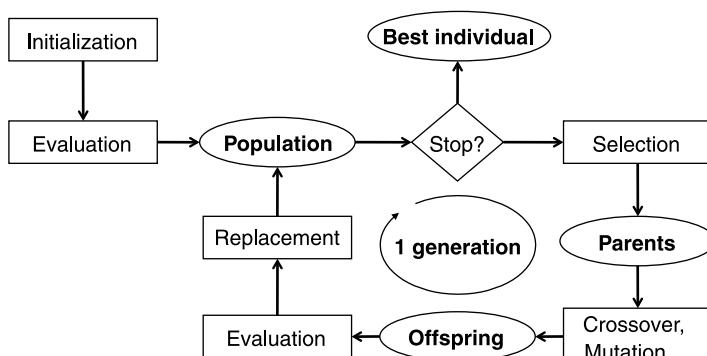


Figure 2.9

General scheme of evolutionary algorithms.

creates children. Some extra “genetic drift” is generated by the mutation operator, which transforms individuals in a stochastic way. When the breeding and mutating processes are complete, the survivor selection operator is used to select which individuals will form the next generation.

The name “evolutionary computing” is a broad term and has subdomains. Three of these are genetic algorithms (GA), evolutionary strategies (ES), and differential evolutions (DE).

Genetic algorithms were developed in the 1960s and made widely known by Holland in the 1970s (Holland, 1975). By design, this method optimizes problems with parameters that can be represented by finite bitstrings (typically integer parameters). All the operators use a binary representation to act upon; e.g., the mutation operator will flip bits in some of the individuals. GA are widely used today, but since we focus this chapter on methods that are used to optimize real parameter problems, we will not go into more detail. Interested readers can refer to the many textbooks devoted to GA, such as Eiben and Smith (2003).

Evolutionary strategies were introduced by Rechenberg (1973) and Schwefel (1975) in Germany at the same period as GA, but the two fields evolved separately for two decades. They are mainly designed for problems with continuous (real) parameters.

The most important operator in such methods is the mutation operator. It acts on individuals of the population by adding, to the different parameters, random values that follow a Gaussian distribution. So if x_i is one of these parameters, the mutation operator defines x_{i+1} by $x_i + r_i$, with r_i drawn from the normal distribution $N(0, \sigma)$.

The value of σ , the size of the mutation step, will determine the balance between exploitation and exploration. In the simplest case it is fixed, but this option is not very efficient. Dynamic mutation step sizes have been proposed that decrease deterministically with the number of generations, so that exploration is dominant at the beginning of the search and is progressively replaced by exploitation. However, the speed of decrease needs to be adapted to every problem. Based on theoretical consideration, the “1/5 success rule” of Rechenberg (1973) solved this issue by looking at the ratio of successful mutations, i.e., mutations for which the offspring is fitter than its parent(s). If this ratio is greater than 1/5, there is too much exploitation and σ should be increased; if it is lower than 1/5, then σ should be decreased.

Today most types of ES are self-adapting, meaning that σ itself is evolving during the algorithm run. In the simplest form, at the start of the evolution, a random initial value is used and then in every generation the next σ' is mutated with the following formula: $\sigma' = \sigma e^{\tau N(0, 1)}$, where τ is an arbitrary constant. Depending on the complexity of the problem to be solved, you can also use mutations with a different step size for every parameter. The most sophisticated mutation operator is actually the correlated mutation, where every parameter has its own step size and a correlation matrix is calculated to rotate the mutation ellipse toward the steepest descent. Of course, this

comes at the cost of an increasing number of parameters to be evolved since all the σ_i as well as all the correlation coefficients have to be adapted. Readers interested in how the matrix is calculated and adapted can refer to Eiben and Smith (2003). There is no requirement to decide whether to use a single step size, multiple step sizes, or a correlated mutation. The simplest is probably the fastest, but if it does not converge to a good solution, then it might be useful to consider more sophisticated mutation paradigms.

At the origin of ES, populations were made up of a single individual and no recombination was used, mutation being the only operator responsible for evolution. Now it is common to use larger populations. For every individual offspring, parents are chosen at random in the previous generation. The parent selection, as the survivor selection, can follow different strategies that we will detail later. In contrast with real natural evolution, it is possible that an offspring will be created from more than two parents recombined with each other.

Several methods are available for the recombination operation. A first distinction is made between local and global recombination. Local in this context means two or very few parents. This has as an effect that the offspring is a combination of a few possible solutions and stays local. The opposite method is a global recombination in which all the parents are used to create one individual. The recombination operation can also be “discrete” or “intermediate.” In the discrete case, for every parameter a separate choice is made to decide from which parent the parameter will be inherited, and this parameter is copied to the child; e.g., if there are two parents x and y with parameters $(x_1, x_2, x_3, x_4, \dots)$ and $(y_1, y_2, y_3, y_4, \dots)$, a possible child would be $(x_1, y_2, y_3, x_4, \dots)$. The other way is the intermediate recombination, which creates children of the form $[(x_1 + y_1)/2, (x_2 + y_2)/2, (x_3 + y_3)/2, (x_4 + y_4)/2 \dots]$. Often the model parameters are recombined in a discrete way to increase diversity in the offspring, while for the algorithm parameters (e.g., the σ_i), an intermediate recombination is used.

The selection operator in evolutionary algorithms can be fitness proportional or rank based. In the latter case, the rank of an individual in the population is used as criterion for survival. Several methods are commonly used. A “roulette” selection scheme works like a roulette in a casino; all individuals get a number of slots on the roulette according to their fitness value. To select N individuals, the roulette is spun N times and the slots in which the ball lands are selected. Another popular method is called “tournament” selection. As the name implies, a small number of individuals randomly chosen from the population participate in a tournament; the one with the best fitness value wins and is selected to survive.

One also has to decide from which group of individuals the survivors are selected. In the world of ES, if it is among the parents and the offspring together, this is called

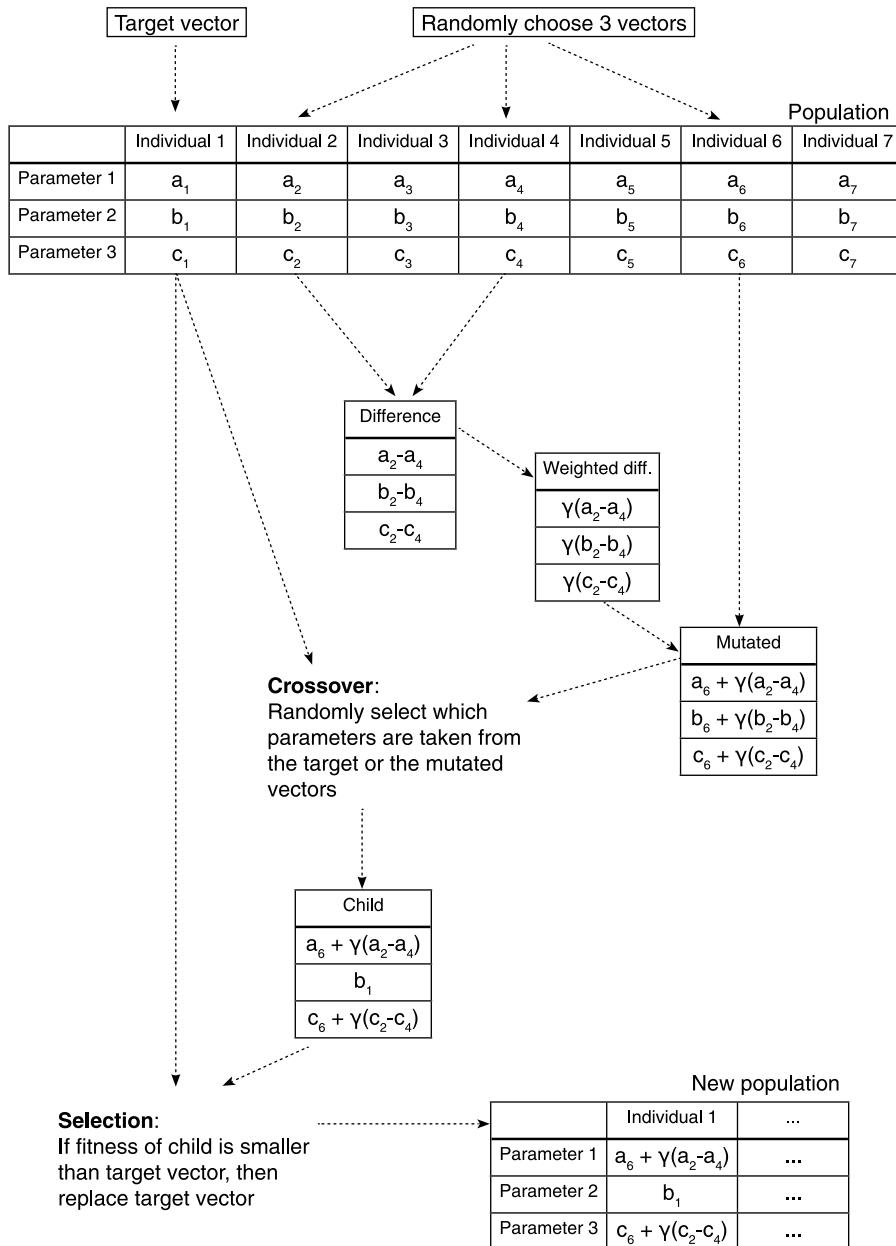
a plus strategy. The opposite method is the comma strategy, where the parents die off and only some of the offspring survive. A special notation shows which strategy is used in the algorithm. If one uses a (10+20)-ES, it means that every generation consists of ten parents and twenty children and that the plus strategy is used during the selection. A (10,20)-ES will refer to the same population with a comma strategy.

Differential evolution is a method similar to ES since it is an evolutionary algorithm for problems with floating point parameters. It was proposed for the first time in 1995 and published in 1997 (Storn and Price, 1997). In this case, all individuals of a generation at a certain time can be the parent of exactly one new child. To create this new individual, a random selection takes place that decides which parameters will be mutated, the other parameters being inherited without any change. Next, three other individuals (x , y , and z) different from the parent are randomly selected. For every parameter j to be mutated, the new value for the offspring is $n_j = x_j + \gamma(y_j - z_j)$, where γ is a positive real number. By doing this, the mean size of the mutation, $y_j - z_j$, becomes totally dependent on the previously generated points and the algorithm does not need any extra parameters to tune its size. The fitness function of this new individual is calculated and the child will replace the parent if its value is better than that of the parent. This whole procedure is illustrated in figure 2.10. Interested readers should refer to Price et al. (2005).

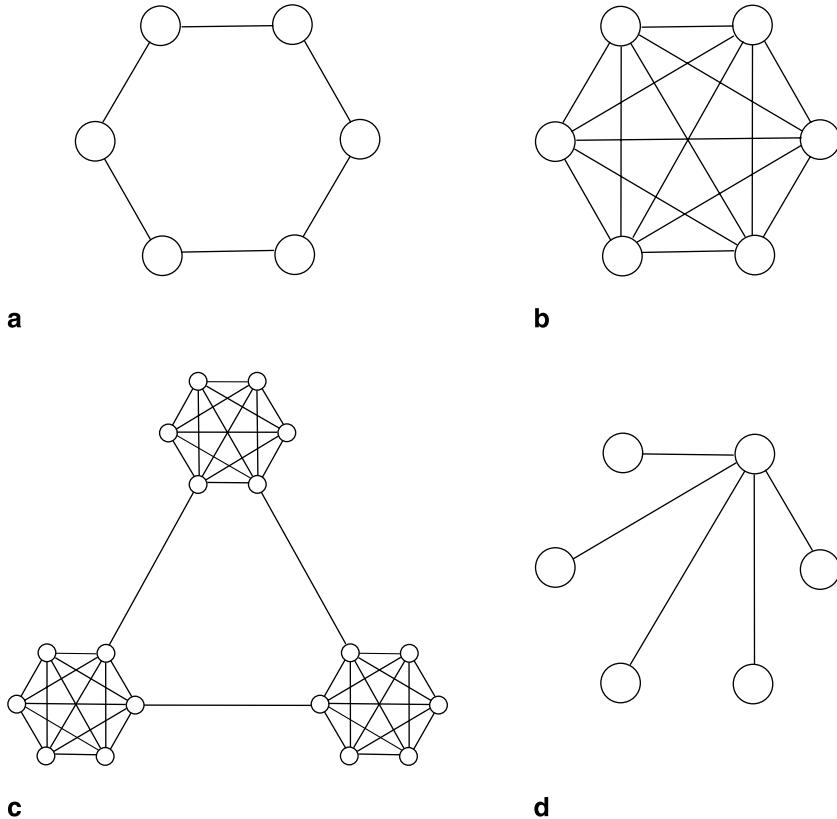
Evolutionary algorithms are well-established methods, with a large background of experimental and theoretical research supporting them. They are very flexible because they can be used in almost any situation where global search techniques are applicable. Parallelization is easily implemented by running the fitness evaluation of different individuals of one generation on separate central processing units (CPUs). A disadvantage is that a lot of algorithm parameters can be tuned and it is not always obvious how changing one parameter may affect the performance of the algorithm. ES and DE generally outperform GA for real-number problems because they can reduce the size of their mutation with time whereas GA can only reduce the probability that a mutation will occur but not the strength of this mutation. DE is simpler, more robust, and often faster than ES but seems not as efficient for noisy optimization problems (Vesterstrom and Thomsen, 2004; Krink et al., 2004).

Particle Swarm Optimization

The particle swarm optimization (PSO) technique, developed in the 1990s (Kennedy and Eberhart, 1995), makes use of the way animals like bees, fish, or birds swarm around in search for food. The general principle is that every individual in the group searches for an optimal solution, making use of local information found in its direct environment and also more global information provided by neighboring individuals. The algorithm starts with the swarm individuals randomly spread over the entire

**Figure 2.10**

Replacement of an individual with differential evolution.

**Figure 2.11**

Examples of swarm topologies. The particles of the swarm can be linked following a ring (a), a star (b), three clusters (c), or a wheeled social structure (d).

solution space. All of these individuals are connected according to a certain topology, in the sense that neighbors in the topology can share information about the solution they have found so far. Examples of such topologies can be seen in figure 2.11, the easiest being the fully connected topology, whereas an organization like that in figure 2.11c creates separate groups inside the swarm. In certain versions of the PSO, particles are randomly connected and the topology changes during the run.

The main principle of PSO is to adapt the position of each particle at every time step according to the rule $x_{k+1} = x_k + v_k$, where v_k is the particle's velocity that changes according to $v_{k+1} = wv_k + r_1c_1(x_{\text{globalbest}} - x_k) + r_2c_2(x_{\text{localbest}} - x_k)$, with c_1 and c_2 positive constants and r_1 and r_2 random numbers. $x_{\text{localbest}}$ is the best solution so far found by the particle, while $x_{\text{globalbest}}$ is the best solution so far found by the particle and its neighbors.

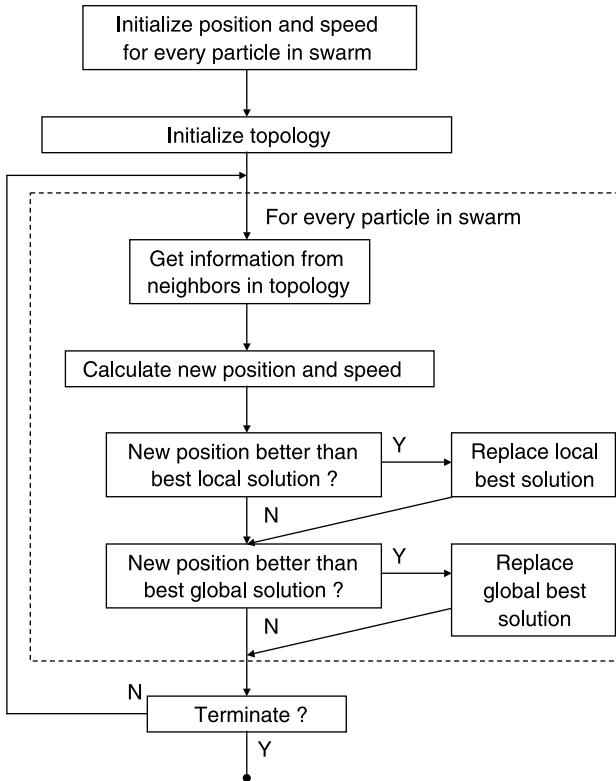


Figure 2.12
The particle swarm optimization algorithm.

The first term creates some inertia of the particles; w is called the inertia weight and is usually set to a value slightly below one. Once an individual is moving in a certain direction, it will try to follow the same direction; this is the exploration part of the method. The second and third terms, on the other hand, are used by the particle to exploit information gathered by itself and its neighbors. Changing the w value changes the balance between exploitation and exploration.

The full PSO algorithm is illustrated in figure 2.12. As with most global optimization algorithms, there are a large variety of ways in which this algorithm can be adapted to problems. Possible changes include the distribution of the initial velocities, the calculation formula for the new velocities, the topology of the individuals, the swarm size, or dynamic changes of w (Engelbrecht, 2005). A “ready-made” version, the Standard PSO, is redefined every year by researchers in the field and can be found at <http://www.particleswarm.info>.

2.5 Dealing with Greater Complexity

In our discussion so far, for the sake of clarity we have described only generic algorithms. These give a better understanding of the philosophy behind each algorithm family. Of course the problems that engineers, scientists, analysts, managers, etc. have to face are not as simple as the test cases we describe. In the following pages we consider more complex problems and how they are handled, without entering into detailed descriptions of the machinery.

Multiobjective Fitness Functions

In everyday life we often deal with problems that have contradictory requirements, such as buying a computer that is both powerful and cheap. The same can occur in computational neurosciences: we may want to increase the number of connections between the neurons of a network while decreasing the total wiring length, or to enhance both learning sensitivity and robustness to noise of a synapse. The fitness functions used in these types of problems are called multiobjective and use the concept of *dominance*.

Comparing two solutions, dominance exists if one solution is as good or better than the other one for every objective and is better for at least one objective. A set of solutions, as shown in figure 2.13, will therefore have some solutions that are dominated and some that are not. The latter, called *Pareto optimals*, delineate a Pareto front. If the user needs a unique solution to a problem (such as buying only one computer) he or she will have to choose among this set of Pareto optimal solutions

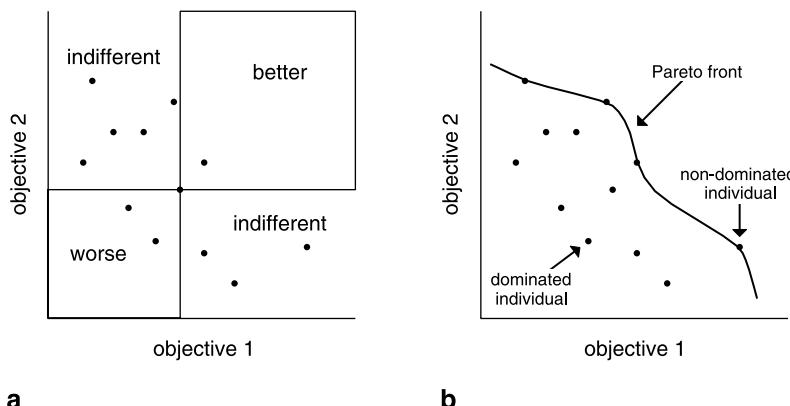


Figure 2.13

A two-objective example of Pareto dominance. The central point in (a) dominates points in the “worse” region, is dominated by a point in the “better” region and is indifferent to points in other regions. The points that are not dominated form a Pareto front (b).

or find an additional criterion to rank the solutions (“I also want it to be light in weight”). However, the goal of a multiobjective optimization algorithm is only to find the Pareto front.

The global algorithms we have discussed in the previous sections require a unique value of the fitness function in order to work. Several strategies have been developed for multiobjective problems. They are separated into three main classes: aggregation-based, criterion-based, and Pareto-based fitness assignments. Those in the first class aggregate the different objectives into a single fitness value, generally with a weighted sum of the objectives, and vary the weights with time. Those in the second class choose one or another objective for each step or during the selection process. For example, parents in evolutionary algorithms can be chosen partly from among the best individuals according to one objective and partly from the best according to another objective. The strategies in the last class use the dominance concept to define fitness; for example, by counting the number of individuals one dominates or is dominated by.

The preservation of exploitation and exploration is more complex in such problems, and readers interested in finding efficient multiobjective algorithms can refer to Deb (2001). Such algorithms have been used in computational neuroscience by Druckmann et al. (2007).

Noisy and Dynamic Fitness Functions

Several phenomena are expected to provoke changes of the fitness function. They mainly belong to two categories: noise and dynamic behavior. Noise is inherent to any experimental measure but also appears when rounding is done in numerical calculations. To some extent, tricks commonly used to reduce computation time, such as approximating a heavy fitness function by using a simpler one, can also be considered as noise. If the signal-to-noise ratio of a model trace is high, fitness functions should not be much affected and the noise is generally neglected. If not, several ideas have been tested to reduce its impact.

The first and more obvious is to resample the solutions, i.e., to make several fitness evaluations for each point and to average them so that the noise is reduced. Another solution, quite similar in philosophy, is to calculate the fitness value of a point by using interpolations from neighboring points. For population-based algorithms, such as evolutionary algorithms or particle-swarm optimizers, an alternative can be to augment the population size. That reduces the effect of noise on the global behavior of the population and hence on its results. Note that all these methods require additional fitness measurements and can be time-consuming. Finally, for algorithms with elitism, such as simulated annealing or plus strategies in ES, thresholding has sometimes been adopted: a new point replaces an old one only if the fitness difference between the two is higher than a certain threshold of size similar to the noise level.

Whichever solution you prefer, if you expect a high level of noise, you should always take into account its impact when you select a fitness function and an optimization algorithm.

Dynamic fitnesses exist when a landscape evolves with time. They are typically expected for embedded systems. In our field, one can imagine a model connected to an active brain tissue with both interacting in a real-time feedback loop; brain-machine interfaces (Lebedev and Nicolelis, 2006) or dynamic clamps (Prinz et al., 2004b) are good examples. In such a system, the best solution at a given moment becomes no longer optimal after a while. At present, the main solutions to this kind of problems are coevolution, exploration, memorization, and explicit actions. With population-based algorithms, coevolution consists of having separated populations evolving in parallel with usually no communication among them. A variation, called niching, enforces separation of the populations by repulsion mechanisms. The idea is to populate several local minima because the global minimum at a certain time is likely to have been a local one previously. More generally, one can enforce more exploration than in standard algorithms, to keep a larger view of the fitness landscape. Memorization strategies try to save information on points already visited so as to react rapidly to changes in the fitness landscape. This is particularly efficient when the global minimum goes back and forth between a few positions. Explicit actions to temporarily increase the diversity, such as augmenting the mutation rate in GA, can also be made at the moment when the fitness landscape changes.

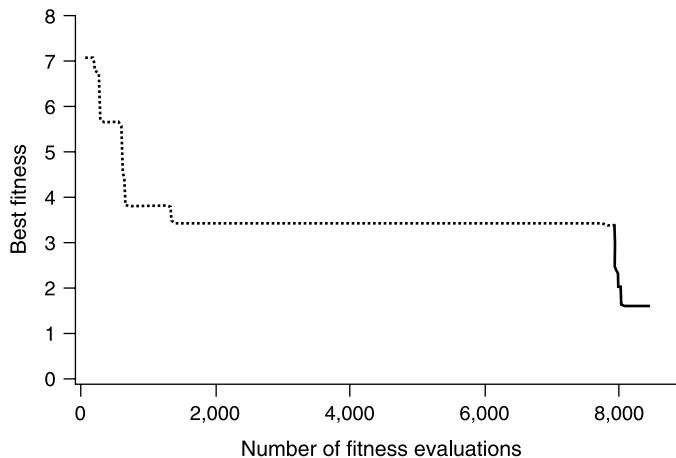
The research field of time-changing fitness optimization is rapidly evolving and the interested reader might want to follow recent development in workshops like EvoSTOC or EvoDOP (<http://www.aifb.uni-karlsruhe.de/~jbr/EvoDOP/cfps.html>).

Hybrid Parameter Search Methods

The term hybrid methods is used for any combination of several optimization techniques. Let us distinguish between simultaneous and consecutive ones.

In the first case, different methods are used simultaneously. Several reasons can favor this approach. For example, one can have a problem with both real and integer parameters. Or one may want to optimize the parameters of an algorithm, such as the mutation rate of a GA, using another algorithm. These kinds of hybrid methods are usually complex. They take long to develop but may be a very efficient solution for a particular problem.

The second class of hybrid algorithms applies one algorithm after the other, sometimes several times. This technique may be practical for switching between exploration and exploitation at different stages of a search. The simplest and most widely used hybrid technique in this category consists of first using a global population-based search method until some stopping criterion is met, and then using a local search algorithm starting from the best point(s) found so far. This method has

**Figure 2.14**

Example of hybrid algorithm performance. After using an evolution strategy for $\sim 8,000$ evaluations (Achard and De Schutter, 2006), a local search algorithm is applied to the best individual of the population. The fitness clearly follows two successive improvements.

proven its efficiency for several biological problems, as is illustrated in Rodriguez-Fernandez et al. (2006) or in figure 2.14.

Constraint Handling

The parameters of the problems we have met so far were unconstrained or, at most, simply bounded by a maximal and a minimal allowed value. However, some problems have more complex constraints, such as inequalities on combinations of the parameters. Here we consider how to take into account forbidden regions of the parameter space.

Most algorithms have simple rejection strategies that eliminate individuals belonging to forbidden zones. This has the disadvantage of not exploiting information from these zones, and rejection becomes a problem if the allowed region is not a continuous space. Indeed, if the global minimum is located in a region that is not initially populated, then the rejection strategy prevents an individual in another region from migrating into the best one using a path passing through a forbidden region.

Some algorithms add a penalty function to the fitness function: forbidden regions can be explored, but their points are unlikely to be attractive since they will have a higher fitness value. But then the algorithm becomes sensitive to the penalty function. If it is too low, the “forbidden” regions are no more forbidden. If it is too high, this method becomes equivalent to the rejection method. One solution is to have a dynamic penalty function that increases with time so that the algorithm can explore the whole space in the beginning and is later restricted to the authorized region.

There are still some cases for which this method does not work; for example, when the global minimum of the unconstrained parameter space is distant from the global minimum of the constrained one.

Other strategies are sometimes followed, of which some try to separate the fitness and the constraints, for example, with multiobjective algorithms or by a random balance between the two, as in the stochastic ranking ES (Runarsson and Yao, 2000). For a complete review of techniques used to handle constraints in EA, readers can refer to Coello Coello (2002).

2.6 Conclusions

In this chapter we have seen many different ways to automatically tune parameters of a model. “Very nice,” you probably think, “but which one is the best? Which one should I use?”

We have presented several algorithms and fitness functions because none is perfect, despite the hope of their authors. In the world of optimization, a famous theorem, called the no-free-lunch theorem, developed by Wolpert and Macready (1997), says that “all algorithms that search for an extremum of a cost function perform exactly the same, when averaged over all possible cost functions.” In other words, an algorithm that is optimal for one problem will be poor for other problems, as illustrated in figure 2.15.

That said, we can offer some advice. First you should choose your fitness function carefully. If you can minimize the number of local minima or the sensitivity to noise, you will greatly reduce the difficulty, hence the time to find the global minimum. Among the qualities of a good fitness function we can cite minimum waste of information, robustness, simplicity, minimum variance, and minimum use of computer

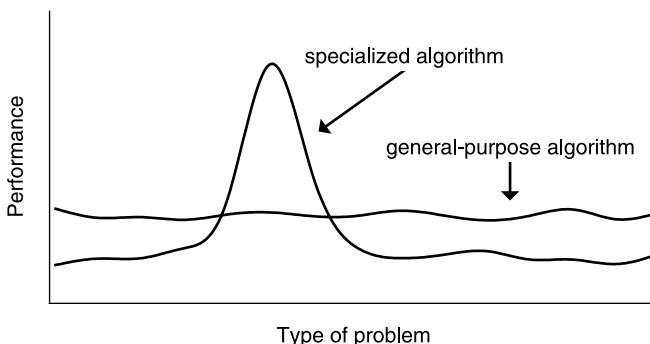


Figure 2.15

Illustration of the no-free-lunch theorem.

time. Combinations of different fitness functions are always imaginable, as in Keren et al. (2005) or in Weaver and Wearne (2006).

Second, if the no-free-lunch theorem states that no algorithm outperforms others for *all* problems, it also clearly states that for *your* problem some algorithms will do better than others. Therefore you should start with a list of properties of your problem: real or integer numbers, noisy or not, constrained or not, with a smooth or complex fitness landscape (Achard and De Schutter, 2006), with one or more objectives, with few or many dimensions, etc. You should also list your resources. Do you have access to a single personal computer or a large cluster of computers? Are you able to use a Fortran or a C++ code or do you need ready-to-use software? Is your laboratory neighbor an optimization expert? Do you have days or years to develop your model? With such lists, it is likely that your choice of an algorithm will be much reduced.

Third, if you have a problem that requires a lot of computing power and weeks of calculations, then it may be worth trying different algorithms, different fitness functions, and different algorithm parameters. For compartmental single-neuron models, Vanier and Bower (1999) demonstrated, for example, that genetic algorithms and simulated annealing are more efficient than conjugate-gradient descent or random searches and that GA were more suited for complex models whereas SA was faster for simpler ones. For a biochemical pathway with thirty-six free parameters, Moles et al. (2003) have tested several recent algorithms, including DIRECT, differential evolution, and different evolution strategies, and found that stochastic ranking ES (Runarsson and Yao, 2000) was the most efficient one for their problem. Doing similar performance tests will not only give you the best algorithm for your own problem but also a deeper knowledge of its mechanisms. But be aware that comparisons published on test cases are often a matter of debate since each algorithm tested generally performs quite differently with different internal parameter settings. Of course, stochastic methods should always be tested with several random seeds.

Finally, remember that if an algorithm specialized for your problem, with internal parameters properly set, will certainly outperform all others, the improvement in performance should be balanced against the development time required. Hence if you have a quite simple problem to solve, you should not invest many days in improving your algorithm's performance by a few minutes or hours. Among the qualities of a good fitness function and optimization algorithm, one should always consider a minimum loss of a scientist's time. Thus ready-to-use solutions are often advised. We do believe that Neurofitter, the software package we recently developed, is an easy-to-use solution for most of the problems intended to reproduce a given electrophysiological activity. It automatically computes phase-plane trajectories and includes several optimization solutions that are well suited for real-number tuning (see the software appendix at the end of this volume).

2.7 Addenda

The Chi-Square Test

The Pearson's chi-square test is a statistical hypothesis test. It tests a null hypothesis that the experimental sample of an unknown distribution is consistent with a given theoretical distribution. A *p*-value is calculated by comparing the χ^2 value with its expected distribution.

As a rule of thumb, if the errors are normally distributed, then a good model should have a $\chi^2/\text{d.o.f.}$ value close to one, where d.o.f. is the number of degrees of freedom, equal to the number of data points (this is 12 in the example of figure 2.1) minus 1 minus the number of free parameters in the model (here 1, k). The curves of figure 2.1 have values of $\chi^2/\text{d.o.f.}$ equal to 3.7, 0.8, and 1.3, respectively. The first one can be considered as "bad" and the last two as "good enough."

Although we must discard models with values of $\chi^2/\text{d.o.f.}$ that are too high, values too close to zero are not satisfactory either. Four hypotheses should be considered as causes for a low value of $\chi^2/\text{d.o.f.}$ It can be statistical luck. It can imply that the errors are not normally distributed and therefore that the chi-square test is not relevant. Very often, it is a sign that the errors were overestimated. Finally, it may signify that the model overfits the data. For example, in figure 2.1b, the same data points are fitted with a polynomial of degree 9. Its $\chi^2/\text{d.o.f.} = 0.31/(12 - 1 - 9) = 0.16$; it gives too much importance to small variations and is therefore not of great interest.

Random Number Generators

Stochastic algorithms rely heavily on the generation of large amounts of random numbers. While there are devices that generate "true" random numbers using physical phenomena, such as nuclear decay, in most cases pseudorandom number generators (PRNG) are used to algorithmically produce numbers that have a high degree of apparent randomness. It is important for stochastic algorithms that the PRNG used is of a high enough quality; otherwise undesired biases can be introduced in the output data of the algorithms.

There are several criteria to check for when choosing an appropriate PRNG for a certain problem. Every pseudorandom number generator (that uses a finite amount of memory) will have a period. This means that if you run the algorithm long enough, at a certain point the sequence of numbers already generated will restart. Luckily for most of the modern PRNGs, this period is long enough for practical applications, but it is advisable to check the period of the random number generator before using it to make sure you won't be generating more numbers than this period in your application. Another problem with PRNGs can be a bad dimensional

distribution of the generated numbers. An example is the once widely used linear congruential generator (LCG), for which an n -dimensional plot of several points created by the algorithm shows $(n - 1)$ dimensional planes in the space, demonstrating that the generator doesn't produce uniformly distributed numbers and is therefore flawed.

We strongly suggest that people never blindly trust PRNGs that are provided by third-party software (libraries). In the past there have been numerous examples of low-quality PRNGs that were often used only because they were part of a widely used software distribution. An infamous example is the RANDU generator that was available for decades on IBM mainframes, and which made many of the results that were produced using that generator unreliable. Also, the built-in PRNGs of the C and C++ languages (*rand* and *srand*) are often badly implemented and should definitely be avoided (Press et al., 2007).

For a general user, it might be difficult to judge whether a certain random number generator is of good quality. A possible way out of this is to perform (a combination of) statistical tests for random number generators, e.g., the DIEHARD (Marsaglia, 2003) and TestU01 (L'Ecuyer and Simard 2007) tests. Implementations of these are freely available on the Internet. The best way is to get advice from a person who specializes in random number generators, and who can suggest a PRNG that is suitable for your application.

As a general-purpose pseudorandom number generator, we recommend the Mersenne twister method (Matsumoto and Nishimura 1998). It has been proven to avoid many of the problems PRNGs in general face. Its advantages are that it is fast; its source code is available in different programming languages; and it generates high-quality random numbers suitable for most applications (except cryptography). Other examples of good PRNGs are available from Press et al. (2007).

3 Reaction-Diffusion Modeling

Upinder S. Bhalla and Stefan Wils

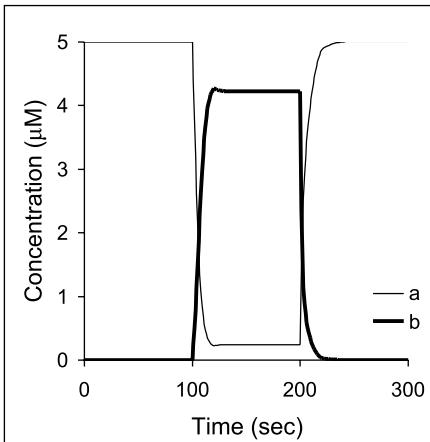
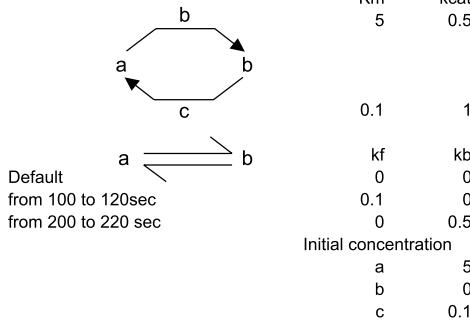
A substantial part of neuronal computation occurs at the chemical, rather than electrical, level. These computations include the many ongoing decisions at each synapse that maintain, raise, or lower synaptic strength. They encompass all the regulatory events that position channels at different places in the cell and assemble or phosphorylate them. Every detail of cellular morphology is an outcome of such computations. Even the cellular housekeeping operations of maintaining cellular ion gradients, adenosine triphosphate (ATP) levels, and protein turnover involve intricate computations. Techniques for modeling such events are a lively area of research and many questions remain unresolved. Several books address these topics, often in depth, with a focus toward cell biological problems (Bower and Bolouri, 2004; Fall et al., 2005). In this chapter we examine the techniques for modeling such processes using increasingly refined numerical techniques. We will also get a glimpse of how to obtain parameters for such models.

The many computational functions mentioned here arise from chemical reactions. In brief, signal identity maps onto chemical identity. Chemical transformations therefore result in signal transformations that have a computational implication. Some examples of computational operations performed by simple chemical circuits are presented in figure 3.1. A key challenge for the field is to integrate the chemical computations that occur at every synapse and dendrite with the biophysical computations that occur at the level of the cell and network. In later sections of this chapter we will discuss how one puts numbers into the reaction models.

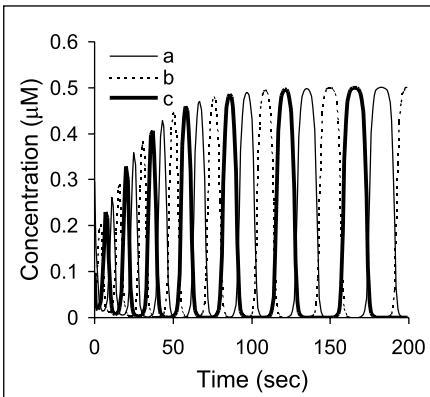
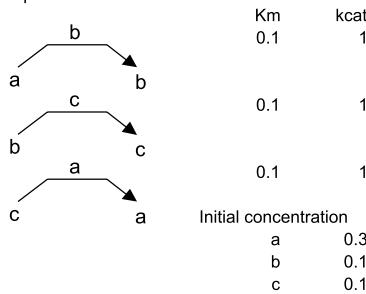
3.1 Stating the Problem

Synaptic signaling is one of the most important intersection points between electrical and chemical computation in the brain. We use an extremely small subset of synaptic signaling events to introduce the techniques we discuss in this chapter. We consider the signaling circuit at four levels: 1, a block diagram; 2, reaction mechanisms; 3, parameters; and 4, solving the model.

1. Bistability. The system can persist in either of two stable states. In one, most of the molecules are in a, and in the other most of the molecules are in b. State change is caused by changing rates of a conversion reaction.



2. Oscillations. This is based on the repressilator circuit (Elowitz and Leibler 2000). Each molecule catalyzes its own formation at the expense of the preceding one in the reaction loop.



3. Amplification. This is based on the MAPK cascade (rates from Bhalla and Iyengar 1999). A small input of active-craf gives a large output of MAPK**

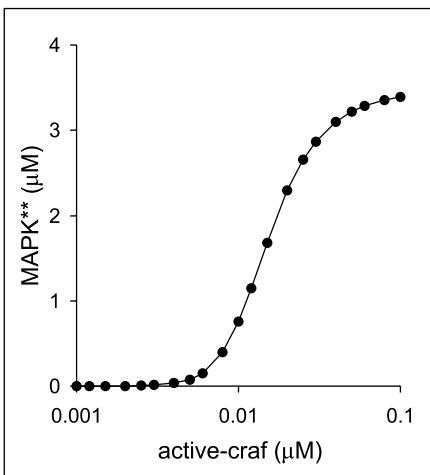
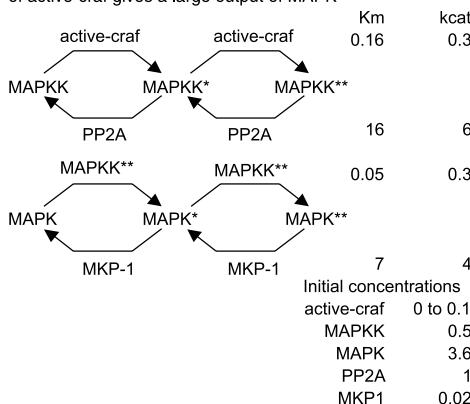


Figure 3.1

Computation with chemical networks. Four examples of computational functions that can be performed by simple chemical networks. These examples are all drawn from biology, although they have been considerably simplified. (1) Bistability, (2) oscillations: repressilator, (3) amplification: mitogen-activated protein kinase (MAPK) cascade, and (4) differentiation.

4. Differentiation. This is a fictional chemical circuit loosely based on receptor desensitization.

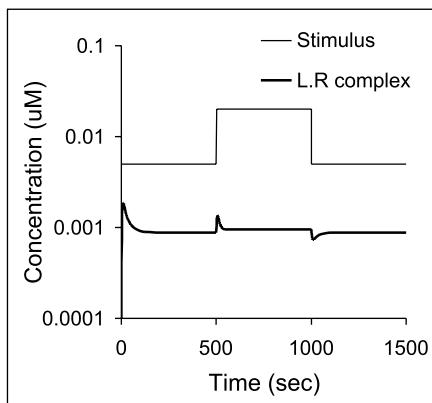
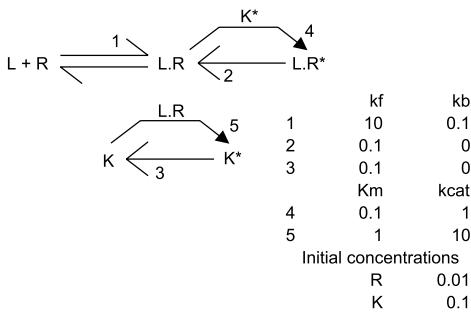


Figure 3.1
(Continued)

Level 1 The Block Diagram

Figure 3.2 is a schematic representation of different steps in the synaptic signaling process and is the most common way in which signaling networks are displayed.

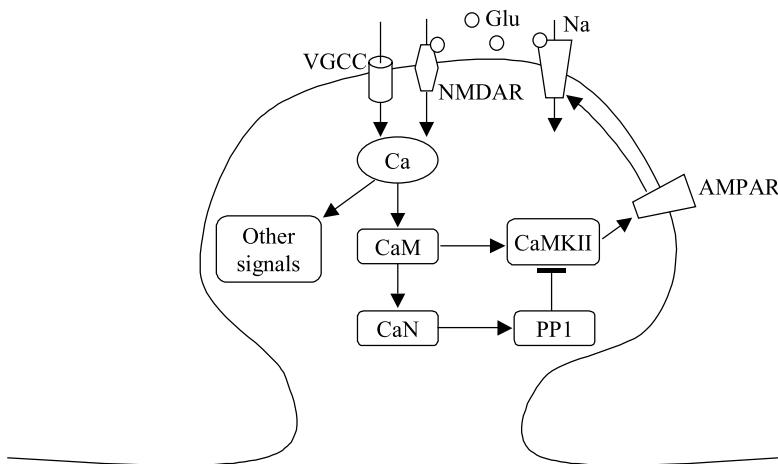
Level 2 Reaction Mechanisms

Here we assign specific biophysical and biochemical mechanisms to each of the arrows in the block diagram. The first arrows represent Ca^{2+} influx through ligand (e.g., N-methyl-D-aspartate NMDA receptors) and voltage-gated (Ca^{2+}) channels. The next set of arrows indicates binding of Ca^{2+} to targets such as calmodulin (CaM). A range of chemical events, including binding, phosphorylation, and dephosphorylation, are represented by further arrows and plunger in this diagram. The overall regulation of α -amino-3-hydroxy-5-methyl-4-isoxazole propionic acid (AMPA) receptor insertion seems clear from the block diagram. However, each of the arrows and each of the blocks in the diagram is deceptive. There are some arrows for ion flow, others for calcium binding, yet others for enzymatic activity, and others for physical movement. Some of the blocks represent single molecule entities such as ions and channels, but others represent complete reaction complexes in their own right. What is worse, each arrow or block could be represented at different levels of detail. Let us consider the activation of CaM by Ca^{2+} , represented in the diagram as a little arrow (note that we omit charge from the equations):



The simplest reaction diagram for this would be



**Figure 3.2**

A block-diagram representation of synaptic signaling. Illustration of a small subset of synaptic signaling. Glutamate (glu) binds to NMDA (NMDAR) and AMPA (AMPAR) receptors and activates the post-synaptic terminal. Ligand (NMDAR) and voltage-gated (VGCC) channels permit ions such as calcium to enter the synapse. In its role as a second messenger, calcium activates many targets, including calmodulin, CaM. This activates calcineurin (CaN) and the calcium-calmodulin-dependent type II protein kinase (CaMKII). CaN activates protein phosphatase 1, which dephosphorylates and inhibits CaMKII. CaMKII phosphorylates AMPA receptors, which move into the postsynaptic density.

However, this simple form does an injustice to a moderately complex reaction. There are two calcium binding pockets on CaM, and each has sites with different affinities. We might then want to write:



This is a bit better. But which calcium binds first? In fact, any of the calcium ions could bind to any of the sites, with possibly different affinities, depending on what was already bound. With four sites that could be occupied, we have sixteen possible states, and thirty-two possible Ca^{2+} binding steps. We could draw this reaction scheme as a four-dimensional hypercube if we really wanted to get elaborate. There are some specialized simulators that specifically address these issues of chemical combinatorics (Morton-Firth and Bray, 1998; Blinov et al., 2004). The key point is that even an apparently simple reaction can often be represented with simple or complex mechanisms. It is an important step to decide which is best for your model.

Level 3 Obtaining Parameters for Some Reactions

Consider the intermediate reaction scheme here, where calcium binds in four stages. How would we parameterize this? Fortunately there are rates and binding values in

the literature (Stemmer and Klee, 1994). Lacking such detailed data, we might have to fall back on a dissociation constant (K_d) and a time course, which would suffice to constrain only one reaction. This would force us to use the first, simpler reaction mechanism. Parameterization and selection of model detail is a key part of all modeling studies. We discuss some of the signaling-specific issues with parameterization in a later section of this chapter.

Level 4 Solving the Model

If we were to take a myopic view of the synapse (figure 3.2), we could begin to solve our synaptic model at this point by using one or more of the methods considered in this chapter. These range from simple ODE models of mass-action chemistry, to three-dimensional stochastic reaction-diffusion methods. However, this would give us a very incomplete model. First, calcium influx is determined by biophysical events: channel openings, ion gradients, and membrane potential. Therefore calcium concentration is a function of electrical activity of the cell and synapse and of electrodiffusion (discussed in section 2 of chapter 4). Second, the biophysical events themselves depend on the outcome of our chemical calculations. For example, the AMPA receptor (AMPAR) conductance and kinetics would be affected by the processes in figure 3.2. We therefore have a mixed system of equations, including the chemical signaling calculations and equations from the biophysical model. How do we link these? One useful observation is that biochemical and biophysical signaling systems typically operate on very different time scales. Chemical events involved in signaling are typically in the range of 100 ms or longer, compared with the 100- μ s scales in biophysics. In many calculations it is therefore sufficient to take the calcium waveform as the only contribution from the biophysics. This works well if we do not care about how chemical signals modify channel properties. We simply precalculate a calcium waveform from the electrical model and feed it into the biochemical model (Doi et al. 2005). This one-way approach lets us develop signaling models at the synapse while incorporating input from an electrical model.

In the general case, we do care that the change in the synapse may affect the electrical network. Here we need to manage bidirectional effects, which will involve running two levels of model: signaling and electrical. Let us first consider events from the viewpoint of the signaling model. We require the calcium waveform generated by the electrical model as a primary input. Possibly we may also need to consider neurotransmitter release acting on metabotropic receptors. As the output of the signaling network, we can represent chemical modifications on channels as changes in the number of ion channels in different states.

From the viewpoint of the biophysical model, we need to map the concentration terms for different chemical forms of the channel molecules to the conductance terms of the appropriate channel states. We may also need to modify the channel kinetics. This could be done by implementing distinct forms of the channel and allowing the

chemical calculations to set their respective conductances. These are not difficult operations, but it does take a long time to run biophysical calculations for biochemical time scales of hours.

A final shortcut is to make a quasi-steady-state assumption: Assume that the biophysical runs will settle instantaneously as far as the biochemical calculations are concerned. Then we can run the biophysical model for various chemical states and obtain mean calcium levels as an output for each state. In other words, we use the biophysical model to fill a lookup table that maps chemical state into Ca^{2+} levels that feed back into the chemical model. Within its limitations, this is an efficient way to construct an integrated model of synaptic plasticity that takes both the chemical and electrical signaling into account.

In this initial section we have explored the challenging topic of multiscale models by considering ways to fit a biochemical model into the larger context of a neuronal or network model. In the remainder of this chapter we will look at how to solve the biochemical model.

3.2 Reaction Kinetics in Well-Mixed Systems

The simplest level of description of cellular chemical kinetics is to treat the cell as a well-mixed system where there are no spatial gradients, and to assume that molecular concentrations can be treated as continuous values. This representation is often extended slightly by including a few well-mixed chemical compartments to represent different cellular compartments, with limited intercompartmental traffic. This level of detail accounts for most current signaling models and is the subject of this section. Later we will consider stochasticity and space, but even these more sophisticated representations build on the chemical descriptions framed at the simplest level.

Mass-Action Kinetics

Chemical kinetics is the starting point for modeling chemical computation. Take the reaction:



The contribution of this reaction to the time evolution of A is described by

$$\frac{d[A]}{dt} = -k_f \times [A] \times [B] + k_b \times [C]. \quad (3.5)$$

The brackets represent the concentration of the named species. More generally,

$$\mathbf{s} = \mathbf{N} \times \mathbf{v}, \quad (3.6)$$

where s is the vector of the rate of change of each molecule, N is the stoichiometry matrix, and v is the velocity of each reaction. The stoichiometry matrix is made by writing out the order of each molecule in successive rows. Each chemical equation forms another column. We use negative values for the left side and positive for the right. For equation (3.4) we have a single reaction, so there is a single column matrix:

$$N = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}.$$

In principle, we can define almost any reaction system in terms of elementary reactions of the above form. Enzymes, for example, are typically represented by the Michaelis-Menten form (Michaelis and Menten, 1913), which is two such reactions, one being unidirectional.



It should be stressed that this is one of the simplest of many ways of expressing the detailed mechanisms of enzymes. That is a large topic in itself (Cornish-Bowden, 2004). For the purposes of most signaling calculations, the simple enzyme representation in equation (3.7) will suffice. More complex representations can be expressed as a combination of simple conversion reactions, so there is no loss of generality.

One can solve the equations for an enzyme explicitly as this series of two reactions. This has the considerable merit that we do not ignore the enzyme-substrate (ES) complex. However, the Briggs-Haldane formulation of the kinetics of this equation is also commonly used:

$$\text{rate} = \frac{d[P]}{dt} = -\frac{d[S]}{dt} = \frac{k_{\text{cat}} \times [E] \times [S]}{K_m + [S]}. \quad (3.8)$$

The terms are related as

$$K_m = \frac{k_2 + k_3}{k_1} \quad (3.9)$$

$$k_{\text{cat}} = k_3.$$

The form in equation (3.8) ignores the presence of the enzyme-substrate complex as a molecular entity. This may fail to correctly represent cases when there is competition between substrates for an enzyme site. It also leads to peculiarities when performing stochastic calculations.

We have discussed rate equations using concentration units. There is an implicit assumption here: namely, that all reactions occur between soluble molecules in

well-stirred systems. We will later consider the issue of spatially inhomogeneous systems. However, there is another complication that comes up even in very simple systems: the presence of reactants on two-dimensional membranes. How can we deal with this combination of two- and three-dimensional reactants in mass-action terms? One commonly employed method is to take the total number of such membrane-bound molecules and compute the effective volume concentration that would result if the same molecules were dispersed in solution. This assumes that the membrane-bound molecules are perfectly accessible to the soluble ones. A similar calculation lets us also deal with purely membrane-limited reactions. Another approach is to think of all reactions as taking place at the level of individual molecules, in which case we use number values instead of concentrations. Now it does not matter whether the molecules are in two- or three-dimensional systems, as long as they can interact. The underlying assumptions of this are perhaps a little cleaner, but this approach introduces another complication. Here the reaction rate units are based on the number of molecules in the system, rather than concentration terms. Such units are extensive, that is, size dependent, which means that you need to scale them every time the system size changes, even if the concentration is the same. Concentration units, on the other hand, are intensive (size independent) and avoid this problem. These issues are really interface and data representation problems. As long as the units are handled correctly, either formalism works just as well to set up the actual mass-action calculations.

Integration Methods

The system in equation (3.6) is a particularly simple system of nonlinear ODEs. In many cases a simple, explicit method, such as an adaptive-time step Runge-Kutta, suffices to compute values with a reasonable mix of speed and accuracy (Press et al., 2007). Frequently, though, the equation system becomes stiff, that is, some reactions are very fast and others are slow. When this happens, the explicit method needs to work at very small time steps to deal with the fast reactions, so it becomes computationally expensive. In such cases an implicit method works better (chapter 1, section 6). One of the best numerical methods, LSODE (Radhakrishnan and Hindmarsh, 1993), actually switches dynamically between methods, choosing implicit methods when stiffness is a problem and going to less complex explicit methods otherwise.

3.3 Stochastic Theory of Reaction Kinetics

In the previous section we looked at reaction systems as continuous rate processes that evolve smoothly over time. In reality, however, reactions are fundamentally based on individual molecular binding and chemical conversion steps. This has two implications:

1. Reactions are really discrete events that involve individual molecules.
2. The occurrence of a reaction depends on the collision of reactant molecules; these collisions are obviously stochastic events that lead to fluctuations.

In reaction systems that contain large amounts of reactant molecules, these fluctuations cancel out and the system's behavior converges to the mass-action kinetics that is adequately described with ODEs. However, for small volumes and small amounts of molecules, this discrete nature makes the behavior of the system inherently noisy. Even large systems in which most species are present in excess, but in which a few key molecules are present in small quantities, can exhibit overall noisy behavior if the activity of these key molecules is amplified by a positive feedback loop. As biologists identify more and more systems in which noise plays an intrinsic role, stochastic descriptions of chemical kinetics become increasingly prevalent.

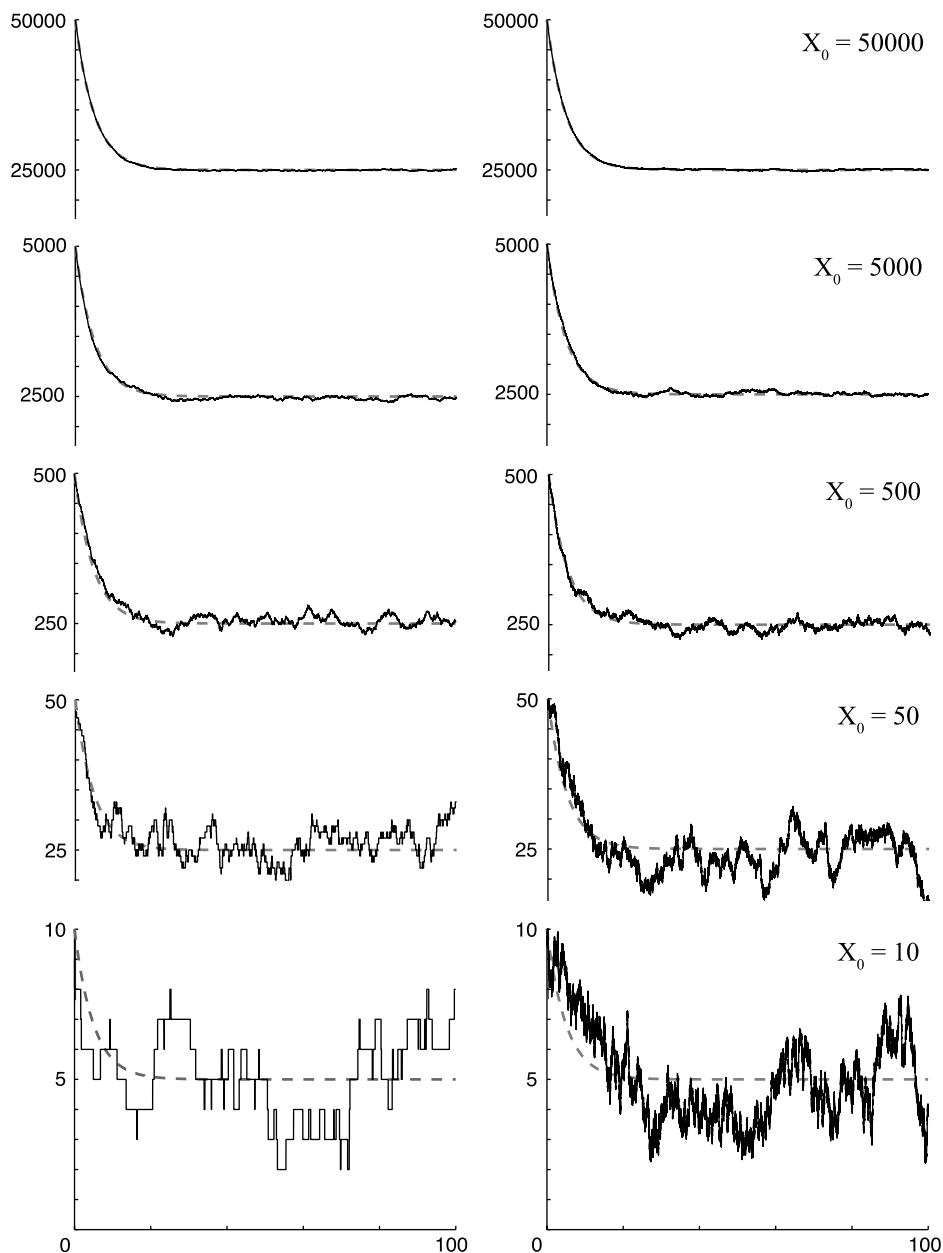
When are stochastic methods applicable? Figure 3.3 shows how the behavior of stochastic systems “automatically” converges to that of deterministic systems as the number of molecules increases. For this reason, it might seem tempting to use stochastic methods by default. Numerical simulation of stochastic models, however, is much more time-consuming and the results can be more complex to analyze. In some situations the choice is clear: a typical synaptic spine has a volume of about 0.1 femtoliters. If the resting concentration of calcium is 100 nM, the spine contains about six free Ca^{2+} ions, which is bound to be stochastic. The point at which it becomes acceptable to use ODEs depends both on the number of molecules and on the reaction kinetics. In practice, any biological reaction involving fewer than 100 molecules is likely to require stochastic methods to describe its behavior.

The Grand Probability Distribution

As in mass-action kinetics, let us consider a well-mixed, isothermal system with some volume V that consists of N chemical species S_1, \dots, S_N , and in which M irreversible reaction rules R_1, \dots, R_M can take place; a reversible reaction can be described by two irreversible ones. To describe a possible state of this system, we will use a population vector $\mathbf{x} = (X_1, X_2, \dots, X_N)$, in which each component $X_i \in \mathbb{N}$ describes the amount of molecules of species S_i as an integer. The overall state of this system at time t is now described by a grand probability distribution $P(\mathbf{x}; t)$ over all these possible states (i.e., the state space). For the simple second-order reaction of equation (3.4), this would be a probability distribution function over \mathbb{N}^3 .

The Mesoscopic Rate Constant and Propensity Functions

In mass-action kinetics, the time evolution of the system was described with a system of ODEs. Here, we need an equation that describes the time evolution of $P(\mathbf{x}; t)$ at a scale where events at the atomic level are averaged into random fluctuations of

**Figure 3.3**

Choosing an appropriate simulation method depends on the numbers of molecules in the reaction system. Consider a single reversible isomerization reaction $X \rightleftharpoons{} Y$. In the left column, we use SSA to simulate this simple system for different initial amounts of X [the black curves represent $X(t)$; $Y_0 = 0$ for all cases; $Y(t)$

individual molecules: the mesoscopic scale. However, before we can come up with such an equation, we must first introduce the *mesoscopic rate constants* c_μ which, for each $\mu \in \{1, \dots, M\}$, describe the probability, per unit of time, that one specific, identifiable pair of reactants of reaction R_μ will collide and react in the mixture. The actual derivation of the mesoscopic rate constant is rather complicated, but we can compute c_μ by appropriately scaling the macroscopic rate constant k_μ (which we know from mass-action kinetics) to work with discrete numbers of molecules rather than with concentrations or molecular densities. If we multiply this mesoscopic rate constant by a combinatorial factor $h_\mu(\mathbf{x})$, which expresses the number of uniquely identifiable pairs of reactants for reaction R_μ if the system is in a given state \mathbf{x} , we obtain the propensity functions $a_\mu(\mathbf{x}) = c_\mu \times h_\mu$. These propensity functions $a_\mu(\mathbf{x})$ describe the probability, per unit of time, that one reaction event of reaction R_μ will occur for a given population vector \mathbf{x} . In other words, given that the system is in state \mathbf{x} at some time t , the probability that one reaction event of reaction R_μ will occur in the next infinitesimally small time interval dt is given by $a_\mu(\mathbf{x}) dt = c_\mu h_\mu dt$.

Let us clarify this with a few examples. Consider the generic unimolecular transition $X \xrightarrow{k_1} Z$. The propensity function for this reaction rule is expressed as

$$a(\mathbf{x}) = c_1 \times \#X, \quad (3.10)$$

where $c_1 \equiv k_1$ because k_1 already has units of s^{-1} and expresses the probability per unit time that one molecule of species X will undergo transformation. Multiplying this by $\#X$, which is the amount of molecules of species X given that the system is in state \mathbf{x} , then gives us the probability per unit of time that such an event will occur. For a generic second-order reaction $X + Y \xrightarrow{k_2} Z$ with $X \neq Y$, the macroscopic rate constant k_2 has units $M^{-1}s^{-1}$ and the propensity function is defined as

$$a(\mathbf{x}) = c_2 \times \#X \times \#Y, \quad (3.11)$$

with $c_2 = k_2/V$ and V the volume of the reaction container. The propensity function for a dimerization reaction $2X \xrightarrow{k_3} Y$ is given by the expression:

$$a(\mathbf{x}) = c_3 \times \frac{\#X \times (\#X - 1)}{2}. \quad (3.12)$$

not shown]. The gray curves are the solutions obtained with numerical integration of the corresponding ODEs. At low molecular populations, the SSA solutions show great variability as well as discrete transitions. These features become less and less noticeable as X_0 increases, until the SSA and ODE solutions converge almost fully for $X_0 = 50,000$. The right column shows a similar series of simulations, this time using Euler-Maruyama integration of the chemical Langevin equation [the black curves again represent $X(t)$]. The CLE fails to reproduce the discrete character of the SSA solution for very low numbers of X_0 . However, as X_0 starts to run in the hundreds, solutions obtained with SSA and CLE become increasingly similar. CLE also converges with the ODE solution for very large numbers of molecules.

This time, $c_3 = 2k_3/V$. However, if we take $c_{3b} = c_3/2 = k_3/V$, we still obtain

$$a(\mathbf{x}) = c_{3b} \times \#X \times (\#X - 1). \quad (3.13)$$

As a result, c_{3b} is always used in practice, even though strictly speaking it is not identical to the actual mesoscopic rate constant, but rather to a quantity derived from it. Of course, the remarks on dealing with membrane-bound molecules we made earlier in the section on mass action kinetics also apply here.

The Chemical Master Equation

With the propensity functions defined, we can finally describe how the grand probability function $P(\mathbf{x}; t)$ changes over time through the occurrence of reactions. Let us put the question like this: If the probability distribution over the entire state space is known at some time t , then what is the probability that the system will be in some specific state \mathbf{x} after an infinitesimally small time interval dt ? One possibility is that the system is already in state \mathbf{x} at time t and that no reaction occurs during $[t, t + dt)$. This probability is expressed by

$$\begin{aligned} P_0(\mathbf{x}; t + dt) &= P(\mathbf{x}; t) \left[1 - \sum_{\mu=1}^M a_{\mu}(\mathbf{x}) \right] dt \\ &= P(\mathbf{x}; t)[1 - a_0(\mathbf{x})] dt, \end{aligned} \quad (3.14)$$

where $a_0(\mathbf{x}) dt$ is the zero propensity and expresses the probability that “some” reaction will occur during the next infinitesimal interval dt . The second way to reach state \mathbf{x} at time $t + dt$ is if the system is in some nearby state from which state \mathbf{x} can be reached by the occurrence of one reaction event during $[t, t + dt)$. Since we have M reaction rules that can possibly occur during the time interval, this probability is given by

$$P_{1\dots M}(\mathbf{x}; t + dt) = \sum_{\mu=1}^M P(\mathbf{x} - \mathbf{n}_{\mu}; t) a_{\mu}(\mathbf{x} - \mathbf{n}_{\mu}) dt, \quad (3.15)$$

with \mathbf{n}_{μ} the population update vector for reaction rule R_{μ} . By adding (3.14) and (3.15) and after reordering, we obtain:

$$\frac{dP(\mathbf{x}; t)}{dt} = \sum_{\mu=1}^M P(\mathbf{x} - \mathbf{n}_{\mu}; t) a_{\mu}(\mathbf{x} - \mathbf{n}_{\mu}) - P(\mathbf{x}; t) a_0(\mathbf{x}). \quad (3.16)$$

This equation fully describes the time evolution of the entire grand probability function and is known as the *chemical master equation* (CME). Even though its

name is in the singular, it really expands to an infinitely large system of equations, one for each possible state in the state space. Unless all species in the system have a de facto finite upper bound (in which case the state space can be considered finite and the CME in turns expands to a finite system of differential equations; see Munsky and Khammash, 2006), it should be obvious that dealing with the CME directly will quickly become intractable.

One way of making stochastic kinetics easier to simulate is by looking at the grand probability function as consisting of an infinitely large ensemble of individual sample paths. At each point in time each of these paths passes through exactly one point in the state space. In other words, if we can generate enough sample paths that are each very accurate or even exact realizations of the CME, we can statistically approximate the grand probability function at various points in time.

This is the approach taken by the various simulation methods described in the remainder of this section. Each of these methods depends, in one way or another, on a pseudorandom number generator (chapter 2, section 7). A collection of sample paths can then be obtained by applying a method multiple times to the same model with the same initial conditions, but with different seed values for the random-number generator. Typically, between 100 and 1,000 iterations can provide a good insight into the behavior of a system, but to obtain high-quality data and figures, up to 100,000 iterations need to be performed.

The Gillespie Algorithm

A Monte Carlo method for generating mathematically exact realizations of the CME was introduced by Daniel T. Gillespie in 1976 (Gillespie, 1976, 1977). It is commonly referred to as the *stochastic simulation algorithm* or SSA and can be regarded as the workhorse in the field. The SSA is an attractive choice, not only because of its exactness with respect to the CME, but also because it is simple to implement. The advantage of the SSA, namely, the fact that it tracks each reaction as a separate event, is also the cause of its biggest disadvantage: the SSA can become very slow. Approximate methods are presented in later sections, but for all these methods, the rule applies that solutions obtained with the SSA should be considered as a point of reference.

The key to Gillespie's SSA is the formulation of a joint reaction probability density function $P(\mu, \tau)$ such that $P(\mu, \tau) d\tau$ denotes the probability that given state \mathbf{x} at some t , the next reaction will occur in the time interval $[t + \tau, t + \tau + d\tau)$ and will be of type R_μ . Given our earlier assumptions, the exact expression for this function can be written as

$$P(\mu, \tau) = a_\mu(\mathbf{x}) \times e^{-\sum_{v=1}^M a_v(\mathbf{x}) \cdot \tau}. \quad (3.17)$$

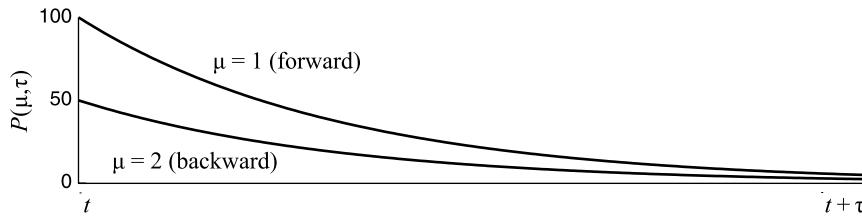


Figure 3.4

The joint reaction probability density function. Consider the simple reaction system defined in equation (3.4). The forward and backward reactions have respective indices of $\mu = 1$ and $\mu = 2$ and we choose their mesoscopic reaction constants to be $c_f \equiv c_1 = 1$ and $c_b \equiv c_2 = 1$. Suppose this system is in state x with $\#A = 10$, $\#B = 10$ and $\#C = 50$ at some unspecified time t . If we plot the joint reaction probability distribution function $P(\mu, \tau)$ for this situation (for $0 \leq \tau \leq 0.025$ s), we can see that it consists of two exponential processes (because $M = 2$ in this case), with the forward reaction being the most probable to occur next [because $a_1(x) = 1 \times 10 \times 10$ is larger than $a_2(x) = 1 \times 50$].

Figure 3.4 plots $P(\mu, \tau)$ for the simple reaction system of equation (3.4). Because $M = 2$ in this system, it basically consists of two exponential processes whose constants depend on the state x . Currently, there are three formulations of the SSA; all are mathematically equivalent and therefore also exact with respect to the CME, but differ mainly in the way they use pseudorandom numbers to iteratively generate (μ, τ) samples from this distribution $P(\mu, \tau)$.

The Direct Method

We start with the direct method (DM; Gillespie, 1976), which is still the most commonly used of these variants. Starting at time $t = 0$ with an initial population vector x_0 , the DM generates a single sample path by applying the following loop until the requested end time $t = t_e$ is reached:

1. For each reaction R_μ , (re-)compute the value of the propensities $a_\mu(x_t)$; also compute the zero propensity $a_0(x_t)$.
2. If $a_0(x_t) = 0$, terminate the loop (because no further reactions can occur).
3. If $t \geq t_e$, terminate the loop.
4. Generate a pair (μ, τ) from the probability density $P(\mu, \tau)$:
 - a. First, we generate two pseudorandom numbers r_1 and r_2 such that $r_1, r_2 \in U(0, 1)$ (see chapter 2, section 7).
 - b. Use the first pseudorandom number r_1 to compute the time step τ at which the next reaction will occur:

$$\tau = \frac{1}{a_0(x_t)} \ln \left(\frac{1}{r_1} \right). \quad (3.18)$$

c. Use the second pseudorandom number r_2 to find the index μ of this next reaction:

$$\sum_{i=1}^{\mu-1} a_i(\mathbf{x}_t) < r_2 \times a_0(\mathbf{x}_t) \leq \sum_{i=1}^{\mu} a_i(\mathbf{x}_t). \quad (3.19)$$

In other words, μ is computed by cumulating over the series of propensity values until we find the interval that encloses $r_2 \times a_0(\mathbf{x}_t)$.

5. Create the new state $\mathbf{x}_{t+\tau} = \mathbf{x}_t + \mathbf{n}_\mu$; update the time $t = t + \tau$ and go back to step 1.

The algorithm can be interrupted after step 5 at regular time intervals to generate output. These interruptions can also be used to apply external changes to the state of the reaction system, e.g., to simulate the sudden injection or removal of molecules. In principle, such interruptions interfere with the sampling process and undermine the exactness of the algorithm. However, if one makes sure that the interruptions occur at a much slower time scale than the chemical events themselves, this will not be a problem.

Finally, the efficiency of the DM can be significantly improved by implementing two simple modifications to the core loop (Cao et al., 2004). First of all, the application of some given population vector \mathbf{n}_μ in step 5 will not change the values of all propensity functions, but only affect the propensity function of the reactions whose reactants are affected by \mathbf{n}_μ . We can therefore avoid unnecessary recomputation of propensity functions by generating an $M \times M$ dependence graph prior to simulation and use this table to quickly look up the subset of $a_{1\dots M}(\mathbf{x})$ that has to be recomputed after selection and execution of some reaction R_μ . Of course, $a_0(\mathbf{x})$ always has to be recomputed. Second, step 4c (in which the index of the next reaction is determined) can be accelerated by periodically reordering the cumulative propensity table from large to small values.

The First Reaction Method

The second version of the SSA, called the first reaction method (FRM), was introduced together with the direct method (Gillespie, 1977). Even though it is rarely, if ever, used in practice, it is useful because it elucidates the nature of the joint probability density function $P(\mu, \tau)$ as being composed of M independent, properly scaled, exponential distributions (figure 3.4). The FRM differs from the DM in step 4 of the loop, where a pair (μ, τ) is generated. Rather than generating two pseudorandom numbers, the FRM requires the generation of M pseudorandom numbers $r_{1,\dots,M} \in U(0, 1)$. We then generate a “tentative” time of occurrence τ_μ (which is relative to the current time t) for each reaction R_μ individually:

$$\tau_\mu = \frac{1}{a_\mu(\mathbf{x}_t)} \ln\left(\frac{1}{r_\mu}\right) \quad \mu = 1 \dots M. \quad (3.20)$$

Using these tentative times, the algorithm generates the pair (μ, τ) by selecting the smallest of these τ_μ .

The Next Reaction Method

Gibson and Bruck (2000) introduced the next reaction method (NRM) as an efficient reformulation of the FRM. The NRM uses an indexed priority queue to keep track of the τ_μ values, which are now absolute values; i.e., relative to $t = 0$ rather than to the last reaction time. At each iteration, the next reaction μ is simply the first element in the priority queue. After execution of this reaction, the propensity values of all (affected) reactions μ' are recomputed as before and new values for their $\tau_{\mu'}$ values are computed using the expression

$$\tau_{\mu'} = \left(\frac{a_{\mu'}^{\text{old}}}{a_{\mu'}^{\text{new}}} \right) (\tau_{\mu'} - t) + t. \quad (3.21)$$

The new time-of-next-occurrence of reaction μ itself, however, is still computed by equation (3.20). This means that the NRM consumes only one pseudorandom number r during each iteration and avoids unnecessary recomputation; it was therefore widely regarded as being the most efficient variant of SSA. However, the indexed priority queue in the NRM is essentially an implementation of a heap, a data structure that is costly to maintain and exhibits inefficient memory access patterns. For this reason, many recent implementations are again based on the (optimized) direct method (Cao et al., 2004).

Leaping Methods

The SSA is exact with respect to the CME because it simulates each occurrence of a reaction as a separate event. This exactness has a computational price, which becomes expensive when the numbers of molecules in the reaction system start to go into the hundreds and beyond. At this point, we may still not want to apply mass-action kinetics, but we also do not really need the full exactness offered by the SSA, at least not for all reactions in the system. As we have seen in the section on SSA and more specifically in the context of the FRM, the next occurrence of each reaction in the system is essentially described by an exponential distribution. What if, instead of iteratively sampling these exponential distributions to decide when the next reaction will occur, we choose a certain timestep τ and find out how often each reaction will “fire” during this “leap”? This idea has led to a class of methods that are collectively known as τ -leaping methods (Gillespie, 2001).

Much like the CME itself, the probability distribution that would describe how often each reaction will fire during a given time leap is computationally intractable because each occurrence of a reaction will influence the others. However, we can obtain a very good approximation by choosing a value for τ small enough that all propensities remain nearly constant during the leap. This criterion for τ is known as the *leap condition* and if it is fulfilled, we can regard each reaction as an independent stochastic process for the duration of the leaping interval. The number of occurrences of a given reaction R_j during the time interval τ is then described by a Poisson distribution with a mean $a_j(\mathbf{x}_t) \times \tau$, and the state of the system after the leap is then given by

$$\mathbf{x}_{t+\tau} = \mathbf{x}_t + \sum_{\mu=1}^M \mathbf{n}_j \times \text{Psn}[a_j(\mathbf{x}_t) \times \tau], \quad (3.22)$$

where Psn is a random deviate from the Poisson distribution (Press et al., 2007).

In contrast to the SSA, which is exact, τ -leaping makes a conscious trade off between accuracy and efficiency. The value chosen for τ should be large enough to obtain a considerable boost in efficiency from leaping over multiple reaction steps, but small enough to satisfy the leap condition. In a group of τ -leaping methods called explicit methods (Cao et al., 2006), the selection of τ does not involve some future state of the system, only its current and previous states, in a vein similar to explicit methods for numerical integration (see chapter 1). Let $\Delta_\tau a_j(\mathbf{x})$ denote the absolute change in propensity of some reaction R_j that results from advancing the system with leap time τ . Satisfying the leap condition means that we apply some upper bound on each $|\Delta_\tau a_j(\mathbf{x})|$. The difficulty now of course is that we need to select τ before we can draw random numbers from a set of Poisson distributions and therefore also before we know how much the propensities will have changed. Cao et al. (2006) solve this by bounding the mean and variance of the expected changes of the propensity functions. This requires the computation of $M^2 + 2M$ auxiliary functions, which can be impractical, especially when dealing with huge and highly coupled reaction systems. To avoid this, Cao et al. (2006) also propose an alternative procedure that directly bounds the relative changes in the populations of all species. To deal with stiff systems more efficiently, implicit formulations of τ -leaping have been considered as well (Rathinam et al., 2003).

An important complication with the concept of τ -leaping is that even if we choose a value for τ that satisfies the leap condition, the numbers of reaction events occurring during this interval are sampled from Poisson distributions. Since the sample space of a Poisson distribution has no upper bound, we can still end up executing the reaction more often than the number of available reactants would allow. This can be avoided by sampling the number of occurrences from a suitable binomial

distribution (Tian and Burrage, 2004; Chatterjee et al., 2005). Alternatively, Poisson τ -leaping can be complemented with SSA for those reactions where the amount of reactants is below a user-defined critical value. Although both approaches are largely similar in performance and accuracy, the latter is conceptually simpler and easier to code (Cao et al., 2005a).

In essence, τ -leaping methods have many attractive properties but remain the subject of active research. The interested reader is therefore advised to check the literature for the most recent developments (e.g., Soloveichik, 2009).

Continuous Methods

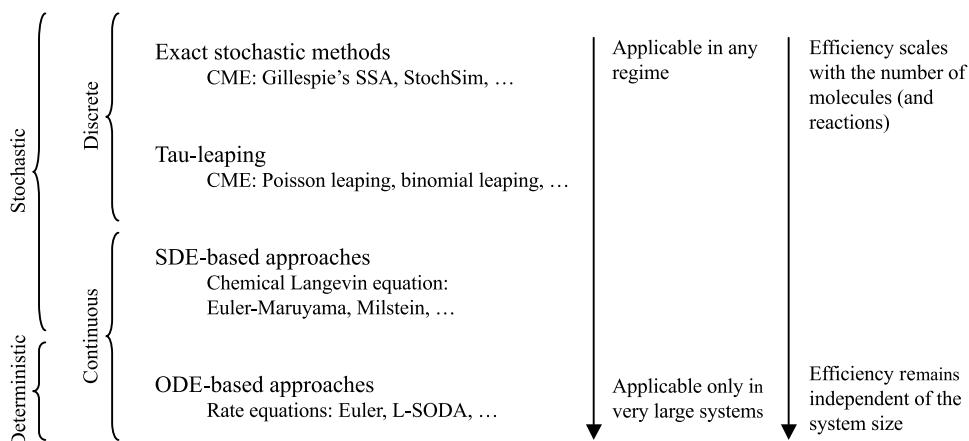
Leaping methods are applicable when all propensity functions remain almost constant during the course of some time step τ . This condition allows us to describe the number of firings of the reactions during the time step τ with a set of independent Poisson distributions with a mean $a_j(\mathbf{x}) \times \tau$. Now assume that each reaction, aside from satisfying the leap condition, fires so often during a time increment that the mean of each Poisson distribution becomes much larger than one. A reaction system operating in this regime will still exert noisy behavior, but its discrete nature is no longer obvious. From a mathematical point of view, the discrete Poisson distribution can be approximated with a normal distribution with a mean and variance equal to $a_j(\mathbf{x}) \times \tau$ and the time behavior system can now be described by a stochastic differential equation (chapter 1, section 6) known as the *chemical Langevin equation* (CLE; van Kampen, 1981; Gillespie, 2000):

$$d\mathbf{x} = \sum_{j=1}^M v_j a_j(\mathbf{x}) dt + \sum_{j=1}^M v_j \sqrt{a_j(\mathbf{x})} \times dW_j(t) \quad (3.23)$$

in which the $W_j(t)$ are M independent Wiener processes (see chapter 1, section 6). Simulation methods based on the CLE are fast because random numbers from a normal distribution are reasonably easy to generate, but also because many reactions can occur during one time step. As was the case with numerical integration of ODEs, multiple explicit and implicit methods have been formulated for integrating stochastic differential equations (SDEs; Kloeden and Platen, 2000; Higham, 2001).

Mixed Methods

Figure 3.5 lists all the approaches we have seen in the previous sections in a hierarchy of convergence. For the smallest of reaction systems, we can only apply the exact methods, such as Gillespie's SSA. Systems that are larger but remain discrete and stochastic can be adequately handled by the approximation of τ -leaping; but we have already seen how τ -leaping is typically combined with Gillespie's SSA to resolve border cases. As the numbers of molecules increases further, the system will lose its discrete character but will still exhibit noisy behavior, at which point SDE-based

**Figure 3.5**

A hierarchy of methods for chemical simulation.

approaches, such as integration of the chemical Langevin equation, become valid. In the thermodynamic limit, i.e., when the volume of the system and the numbers of molecules go to infinity but their ratio remains finite, the system's behavior becomes deterministic and numerical integration of ODEs becomes a valid approximation.

The ordering of methods in figure 3.5, however, is at the same time also an ordering in terms of increasing computational efficiency. Gillespie's SSA is generally applicable but becomes very slow for large systems. Deterministic approaches are valid only for very large systems but invariably are more efficient. This perpetual conflict between validity and efficiency can be avoided to some extent by using mixed methods, in which the overall reaction system is partitioned into a number of subsets, each of which is simulated using a different method.

The issues with these methods are threefold. First, of course, they are always approximate. In addition, stochastic calculations typically assume that molecular numbers do not vary between transitions, and this assumption is violated by the deterministic methods that allow molecule concentrations to vary continuously in time. Second, several problems arise when the same molecules participate in deterministic as well as stochastic reactions in such a scheme. For example, the deterministic (ODE) calculation might give us a fractional molecule count. Third, the different numerical methods need to have some way of aligning time. This is tricky because the best deterministic method as well as the various stochastic methods utilize variable time steps. An important aspect of mixed methods is the approach used to subdivide the reaction system into different regimes. Some methods require user input or an initial subdivision; others perform this subdivision automatically, based on properties of the reaction system.

A pragmatic method for dealing with all these issues was introduced by Vasudeva and Bhalla (2004), in which reactions are automatically partitioned into two subsets, based on the numbers of molecules and their reaction propensities. A stochastic update rule is used for the subset of slow reactions, whereas the fast subset is integrated with a deterministic method (exponential Euler). Burrage et al. (2004) proposed a three-tiered partitioning scheme that classifies reactions into slow, intermediate, and fast regimes, which are simulated with Gillespie's SSA, a form of τ -leaping called Poisson-Runge-Kutta (Burrage and Tian, 2004) and Euler-Maruyama (Higham, 2001), respectively.

Other approaches attempt some type of time-scale separation, so that the effect of a fast, “uninteresting” subsystem is somehow approximated in the coefficients of a reduced CME, which describes only the slow subsystem. These methods are mixed in the sense that the reduced CME is usually simulated with Gillespie's SSA, whereas the impact of the fast subsystem on the reduced system is “simulated” either by some form of stochastic sampling (e.g., stochastic simulation of the “inner” fast subset to establish equilibrium properties) or by theoretical analysis (which effectively eliminates the fast subset from simulation altogether). For instance, Rao and Arkin (2003) proposed to partition the full species vector x into two subsets of “primary” species y and “intermediate” species z , respectively. The partitioning of x into y and z is chosen so that the subset of intermediate species z can be considered to reach a state of stationary quasi-equilibrium on a time scale that is much shorter than the time scale at which the interesting primary subset y changes. This allows us to write a CME for the reduced system:

$$\frac{dP(x; t)}{dt} = \frac{dP(y, z; t)}{dt} \approx \sum_z P(z | y) \frac{dP(y; t)}{dt}. \quad (3.24)$$

The actual shape of this reduced CME, however, must be manually derived on a case-by-case basis, which makes this method not generally applicable. Haseltine and Rawlings (2002) proposed a similar partitioning approach based on the occurrences of reactions, rather than on species, as was the case in Rao and Arkin (2003). The fast subsystem is described by a set of SDEs, whereas the reduced CME describing the slow subset is simulated with a variation of Gillespie's SSA, modified to take into account changes in coefficients resulting from reactions in the fast subset. Variations on these approaches that are mathematically more rigorous can be found in Goutsias (2005), Cao et al. (2005b), and Samant et al. (2007).

3.4 Modeling Spatial Gradients

So far we have largely ignored space by assuming that our reaction system was well mixed at all times. Even though this approach can take us a long way in chemical

kinetic modeling, some molecular mechanisms cannot be fully understood without studying the role of space in more detail. Especially in neurons, cellular morphology and the inhomogeneous distribution of crucial machinery can lead to spatial gradients and diffusive fluxes of critical substances that fundamentally change the dynamics of the reaction system. As in the previous section, we start our discussion of reaction-diffusion systems by focusing on the deterministic case. We first describe Fick's law for "pure" diffusion and then combine this with the theory of section 3.3 to obtain the mathematical tools for describing reaction-diffusion. This is followed by a brief description of ways in which these equations can be discretized and integrated numerically. In the second part, we introduce the concept of Brownian motion, which underlies diffusion, and finish by presenting a number of approaches for stochastic simulation of reaction-diffusion models.

Macroscopic Theory of Diffusion

In section 3.2, a set of stoichiometric rules and reaction constants could be translated, under the assumption of well mixedness, into a system of ordinary differential equations. For a combined reaction-diffusion system, however, the variables that represent the concentration of a certain species are dependent not only on time but also on space. In addition, the concentration of some species at a given location not only changes as the result of local reaction mechanisms but also because of a diffusive flux. Fick's law relates the diffusive flux F_A of some species A to its spatial gradient through a proportionality constant D_A . In the simple case of one-dimensional diffusion, Fick's law is written as

$$F_A = -D_A \frac{\partial A(x, t)}{\partial x}, \quad (3.25)$$

where $A(x, t)$ denotes the concentration of species A at some specific time and location; D_A is the diffusion constant or diffusivity and has units of square meters per second or related units. Its value depends on the ligand, but also on the background medium. If this medium is anisotropic, the diffusivity can be a matrix; moreover, it does not need to be a constant but can be a function of both space and time as well.

Note that the direction of the flux in equation (3.25) is opposed to the gradient, meaning that it seeks to eliminate the gradient over time. Again working in one dimension, the change of $A(x, t)$ over time should obviously be equal to the net sum of mass flowing in (or out) from the left and from the right. Mathematically, this can be expressed with a mass balance equation:

$$\frac{\partial A(x, t)}{\partial t} = -\frac{\partial F_A}{\partial x}. \quad (3.26)$$

Combining equations (3.25) and (3.26), we have found a second-order partial differential equation that expresses the change in concentration of species A as the result of diffusion:

$$\frac{\partial A(x, t)}{\partial t} = D_A \frac{\partial^2 A(x, t)}{\partial x^2}. \quad (3.27)$$

In multiple dimensions, we use the Laplacian operator; for instance, in three-dimensional Cartesian space, equation (3.27) becomes

$$\frac{\partial A}{\partial t} = D_A \times \nabla^2 A = D_A \times \left(\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} + \frac{\partial^2 A}{\partial z^2} \right). \quad (3.28)$$

Combined Reaction-Diffusion Systems

If species A , aside from diffusion, is also involved in reaction mechanisms, we can express the changes caused by these reactions simply by adding local “source-sink” terms. These are of course the terms that we saw in the beginning of section 3.2. For instance, suppose that A also changes because of the reversible reaction rule from equation (3.4). The total change of A over time as the result of reaction as well as diffusion mechanisms is then given by

$$\frac{\partial A}{\partial t} = -k_f \times A \times B + k_b \times C + D_A \times \nabla^2 A. \quad (3.29)$$

We have now used Fick’s law to extend the ordinary differential equations that describe deterministic, well-mixed reaction kinetics to partial differential equations that describe the combined reaction-diffusion process. In practice, this means that mathematical models of reaction kinetics can be augmented to include diffusion; the reaction constants we used earlier for well-mixed models can be reused. What needs to be added to the model are the diffusion constants for all diffusing species. Various experimental techniques exist to estimate these constants (see section 3.7). In addition, describing the initial state of a simulation now requires, not only initial concentrations of species, but also their distribution in space. Finally, analyzing the behavior of reaction-diffusion systems can be more involved than is the case with well-mixed reaction systems; even simple reaction schemes can generate patterns of considerable complexity and beauty when diffusion is added.

Boundary Conditions

Aside from adding diffusion constants and initial spatial gradients, building a reaction-diffusion model also requires us to describe space itself in more detail. It is no longer sufficient to describe our reaction container as merely having some volume

expressed in liters, without any specific shape. More specifically, diffusion can be constrained by the presence of impermeable surfaces, such as the cytoplasmic membrane, or by internal organelles like the endoplasmic reticulum. Generally speaking, we use a Neumann-type boundary condition to describe the absence of a flux through these surfaces. These boundary conditions must be applied to each diffusing species individually and may not apply to all species in a system in the same way; e.g., the small, gaseous, molecule nitric oxide is an important messenger in many signaling pathways. It spreads rapidly by diffusion upon release, but unlike other messengers, it is known to pass through membranes unobstructed. Finally, boundary conditions can also be used to describe reactive surface interactions (e.g., receptor binding) or various passive and active fluxes.

3.5 Integration Methods for Reaction-Diffusion

Similar to how numerical methods for integrating ODEs essentially break up the continuous flow of time into discrete steps, all numerical methods for integrating PDEs must start with spatiotemporal discretization of the mathematical model. This means that they must somehow “decompose” the shape of the reaction volume into a grid or mesh of small nodes or voxels, and then approximate the derivatives in the PDEs and the boundary conditions in terms of these simulation elements to obtain a system of algebraic equations. We can roughly identify two broad classes of spatial grids: structured and unstructured (figure 3.6). The choice of discretization approach and solution method is closely related to the choice of grid. To a considerable extent, all of these choices depend on the level of detail and the coordinate system in which the geometry and boundary conditions are described most naturally. It should not be surprising, then, that we face an even broader choice of approaches here than was the case with well-mixed kinetic models.

Quite often, the relevant geometric features of the cellular morphology can be represented by putting together simple geometric primitives such as spheres, boxes, or cylindrical compartments. If we do not really care about the gradients within these individual compartments, we can treat each compartment as a separate “reaction container” or node, each of which represents a single concentration value for each species. The diffusive coupling between compartments, which is represented by the second derivative in equation (3.27), can then be approximated as an algebraic expression in terms of the concentration values in neighboring nodes. This class of approaches, known as *finite-difference (FD)* methods, is also used for compartmental simulation of electrophysiological neuron models (chapter 11, section 1). FD methods can be simple to code, but since the algebraic expressions can only be formulated for structured grids, they are not well suited for dealing with complicated geometries with varying levels of detail.

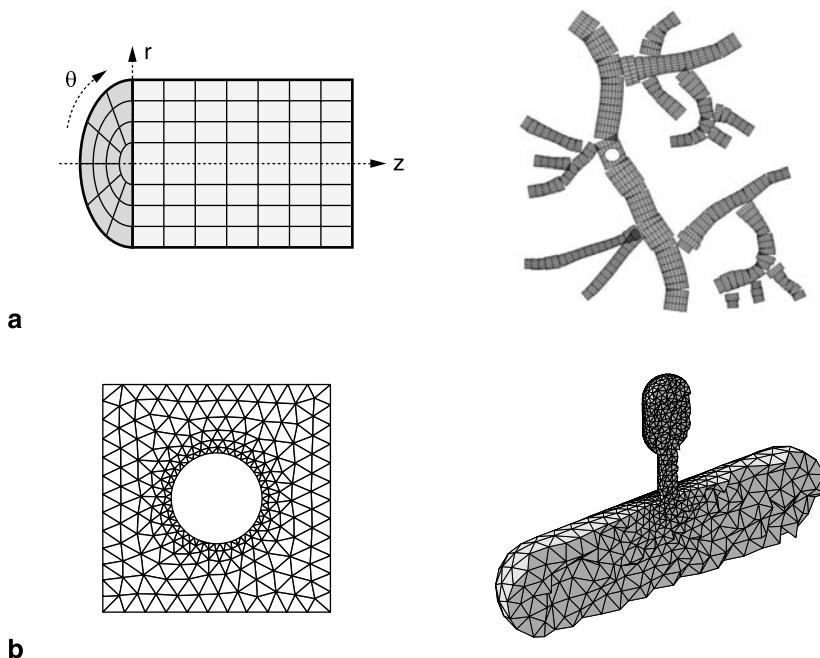


Figure 3.6

Spatial grids for simulating reaction-diffusion systems. (a) Structured or regular grids consist of multiple sets of numbered grid lines. Regular grids are easy to use because grid elements and their neighbors can be referenced with tuples of integer indices. One example of a structured grid is the cylindrical grid (shown at left), which can be used to represent a single stretch of dendritic segment. Grids that are composed of multiple, potentially overlapping structured grids are referred to as block-structured grids. A simple but effective method for performing simulations of reaction-diffusion in detailed reconstructions of dendritic trees is to concatenate multiples of such cylindrical segments through appropriate boundary conditions and intersegment mappings (Santamaría et al., 2006) (shown at right). (b) Unstructured grids can be more difficult to set up, but are best suited for discretizing complex geometries with spatial detail occurring at a wide range of scales. Generally speaking, they work best when combined with finite-volume and finite-element solvers for deterministic simulations, and with voxel-based approaches for stochastic diffusion. For two-dimensional domains, the most commonly used types of unstructured meshes are triangular meshes (shown at left). For three dimensions, tetrahedral meshes are common (shown at right).

A good alternative is formed by the class of *finite-volume (FV)* methods; in these methods, space is discretized into very small volume elements or voxels. The model's quantities are approximated at a center point inside each voxel; the PDEs can then be solved by approximating volume and surface integrals in terms of these central values. Approximations of these integrals can easily be formulated for irregular discretizations (e.g., meshes consisting of tetrahedral voxels), which makes FV methods a good choice for the complex geometries that are typically associated with detailed reconstructions of neuronal morphology. Even though finite-volume methods are easier to understand and program compared with the related but more advanced

finite-element (FE) methods (Ferziger and Peric, 2001), they still require familiarity with a broad range of algorithmic and numerical issues. Their implementation is therefore best left to specialists, and we recommend using one of the existing software packages for fluid dynamics (e.g., FreeFem++/FreeFem3D).

Both FV and FE models make sense when a highly detailed three-dimensional model of cellular morphology is required. The process of defining boundary conditions and preparing them for use in these kinds of simulations is arguably one of the most technically demanding tasks of setting up such a model, typically depending on a wide suite of tools for manipulating 3-D data and models. A good example of this can be found in Means et al. (2006), who describe the various stages required to obtain a computational mesh of the endoplasmic reticulum of a mast cell suitable for simulation of Ca^{2+} dynamics, starting from a stack of images.

3.6 Brownian Motion and Stochastic Diffusion

Earlier, we saw how the molecules in a chemical mixture are continuously bumping into each other and how each reaction is actually the outcome of a collision between two molecules. However, the vast majority of collisions between molecules do not result in reaction events and merely force the molecules to travel along a new vector. Successive collisions give the appearance of a random, erratic path. This pattern of motion is known as Brownian motion and is the microscopic origin of diffusive fluxes. As with stochastic reaction kinetics, it can be useful to simulate this random motion directly rather than simulating smooth, macroscopic changes in concentration values. Tracking the position and velocity vectors of all the molecules in a mixture to compute collision events would clearly be infeasible. Instead, we will describe the displacement of a given molecule during some interval $[t, t + \Delta t]$ as the net outcome of many collisions with a large number of inert “background” molecules present in the solution, such as water molecules. These background molecules are not explicitly simulated; rather, their net effect is implied in the diffusion constant. The motion of a given particle with a diffusion constant D now becomes a simple conditional probability: given that this particle has a known location x_1 at some time t_1 , its location x_2 at some future time t_2 is described by a Gaussian distribution. In one dimension, we have

$$P(x_2, t_2 | x_1, t_1) = \frac{1}{\sqrt{4\pi D(t_2 - t_1)}} e^{(x_2 - x_1)^2 / 4D(t_2 - t_1)}. \quad (3.30)$$

As explained earlier, we do not track collisions explicitly, meaning that this Gaussian applies to each particle independently. In other words, when we are simulating the diffusion of a whole group of particles immersed in a solution, we can apply this conditional probability to all particles independently. Extension to multiple

dimensions is straightforward: when describing the motion of particles in more than one dimension, equation (3.30) applies to each dimension independently as well.

Random Walk Methods

The first category of methods for simulating stochastic diffusion is called random walk methods because they stick very close to the concept of Brownian motion. With these methods, each molecule is stored in memory as a separate object that occupies a certain position in space. At each time step Δt of the algorithm, the particles are relocated to a new position. As described in equation (3.30), we can generate an updated position $x_{t+\Delta t}$ for a particle by sampling pseudorandom numbers from a Gaussian distribution with a mean x_t and variance $2D\Delta t$. Generating samples from a Gaussian distribution during the simulation itself is rather costly but can be speeded up by generating a large number of samples from a normal distribution $N(0, 1)$ once during initialization and applying the following update rule:

$$x_{t+\Delta t} = x_t + \sqrt{2D\Delta t} \times N(0, 1). \quad (3.31)$$

This can be extended to N dimensions by using equation (3.31) to generate N independent displacements. Alternatively, MCell (see the software appendix; Stiles and Bartol, 2001) describes a three-dimensional walk using two probability distributions, one describing a unit-length direction vector and another describing a radial step length r_\perp . MCell generates a large number of samples for both distributions once during initialization and stores them in tables. As a result, MCell only needs to generate two uniform variables for each molecule at each time step, which are used to access these two tables in a random fashion.

These random walk methods do not require a computational grid because they are continuous, which can make it easier for a modeler to set up a simulation. On the other hand, separate routines need to be written to handle various forms of interaction between the diffusing particles and their environment. For instance, in MCell, cellular membranes are described by polygonal meshes. To enforce these boundaries, MCell needs to compute, for each moving particle, whether the step it takes intersects with one of the nearby boundary polygons. If it intersects, MCell must compute whether the particle interacts with particles embedded in the boundary polygon or whether it merely bounces off and continues its step in a reflected direction.

Modeling an interaction with particles not bound to membrane boundaries (i.e., normal reactions in solution) is even more difficult. For instance, the approach taken by Smoldyn, which is a particle-based simulator in many ways similar to MCell, depends on detecting whether the distance between two close-by particles is smaller than a binding radius, which can be derived from Smoluchowski dynamics (Andrews and Bray, 2004). Continuous random walk methods are flexible and accurate but can become slow in practice as a result, especially when a lot of particles are involved.

Random Walk on a Discrete Lattice

A less realistic alternative, albeit one that is simpler to implement, is to overlay the simulation space with a regular, discrete lattice. Molecules can now only occupy “integer” positions x_n with $n \subseteq \mathbb{N}$. Consider the one-dimensional case where the distance Δx between two neighboring lattice points is related to the simulation time step Δt through the relation $\Delta x = (2D\Delta t)^{1/2}$. The random walk for this case can be described with a simple Bernoulli process in which each particle at lattice point x_n jumps to its left or right neighbor with equal probability (Bormann et al., 2001). This will approximate the Gaussian distribution of equation (3.30) in the limit of $\Delta x, \Delta t \rightarrow 0$. The presence of a diffusion boundary between some position x_n and one of its neighbors is handled by adjusting the transition probabilities.

When the number of molecules that occupy a certain position is large, we can improve the speed of this method by letting particles move in (small) groups instead of moving them one-by-one. Aside from the easy implementation of boundary conditions, discrete random walk methods have a second advantage. To deal with mixed reaction-diffusion systems, we can consider the region surrounding each lattice point as a well-mixed volume and apply Gillespie’s SSA (or some other method) to each of these voxels locally in between successive jumps. The main disadvantage is of course our reliance on regular grids. As was the case with finite-difference methods, the lattice needs to be regular, meaning that a rather fine lattice is needed when dealing with complicated geometry.

Voxel-Based Methods

Gillespie himself has suggested that the SSA could be easily extended to include diffusion (Gillespie, 1976). Voxel-based methods for stochastic diffusion start by subdividing simulation space into a number of voxels and are thus conceptually similar to the finite-volume method for solving PDEs. Each of these voxels becomes a separate well-mixed reaction volume, meaning that if the reaction system consists of M reaction rules, and discretization results in N voxels, the Gillespie algorithm has to deal with $N \times M$ exponential processes. In addition, we can now use a reversible “fake” reaction rule to express the hopping of a molecule of species X with a diffusion constant D_X from one voxel with index i to a neighboring voxel with index j :



The “reaction” constants $d_{i,j}$ and $d_{j,i}$ are related to the diffusion constant and depend on the nature of the voxels. For cubic voxels, they are given by

$$d_{i,j} = d_{j,i} = \frac{D_X}{\Delta x^2}, \quad (3.33)$$

with Δx the one-dimensional resolution (i.e., the size of a cubic voxel along its edge). This can also be used with tetrahedral voxels:

$$d_{i,j} = \frac{D_X A_{i,j}}{dx_{i,j} V_i} \quad (3.34)$$

$$d_{j,i} = \frac{D_X A_{i,j}}{dx_{j,i} V_j},$$

with $A_{i,j}$ being the cross-sectional surface area of the interface between the two neighboring voxels, $dx_{i,j}$ the center-to-center distance between the two voxels, and V_i and V_j their respective volumes. A reflecting boundary condition between two neighboring voxels is represented simply by the lack of such a rule.

In contrast to continuous random walk methods such as MCell, voxel-based methods handle both diffusion processes and reaction mechanisms from within a single simulation framework: Gillespie's SSA. This means that they can also benefit from advances in approximate methods such as τ -leaping or mixed methods. Boundary conditions are simple to deal with, regardless of their complexity or the nature of discretization. The main problem with voxel-based methods is that the SSA core loop potentially has to operate on a huge number of exponential processes. MesoRD (see the software appendix), a simulation package operating on cubic voxels, resolves this issue by using an adapted version of the next reaction method called the next sub-volume method (Elf et al., 2003). STEPS (see the software appendix), which is based on tetrahedral voxels, adapts the direct reaction method for large systems by storing the propensity values for each process in a search tree.

3.7 Parameterization

There are four key steps in building a signaling model:

1. Identify key molecules.
2. Identify their interactions.
3. Expand the interactions into chemical steps.
4. Assign rate constants to each step.

If one has spatial information as well, there will be two additional steps:

5. Position molecules and gradients.
6. Specify 3-D boundary conditions for each cellular compartment.

Steps 1 and 2 are the common currency of almost all molecular biology-oriented studies, and the resulting data are richly represented in many diverse databases,

which are listed in HarshaRani et al. (2005). The only tricky part of this stage of parameterization is to decide what not to include. The rule of thumb is to exclude as much as possible. Occam's razor aside, it is simply too hard to keep track of all the interactions, and we have yet to come to the hard part of the parameterization process. Step 3 will require a reasonable understanding of biochemistry and some data about any peculiarities in the mechanism for each enzyme. For example, the simple activation arrow in a typical block diagram:



could mean many things, depending on context. If A is a second messenger, then what the arrow probably means is something like



where B is the active form. Typically there are higher-order terms:



which give rise to computationally interesting nonlinearities. If A is a kinase, the arrow probably expands into



That is, A phosphorylates B, and the phosphorylated state B^* is active. However, if you have phosphorylation, you must also have dephosphorylation:



where the PP is a protein phosphatase. So you will need to find the appropriate phosphatase to reverse this reaction, and it may well turn out to have regulatory implications as well.

In a similar manner, inhibition may be due to binding of an inhibitor, or phosphorylation (again), or some other elementary chemical step.

The simple block diagram arrows can sometimes conceal a vast ignorance. For example, the commonly used arrow:



refers to one of the most complex enzyme regulatory systems in biology. Its mechanism is still incomplete, and these very elusive details are quite central to synaptic signaling. So one must expand those arrows with care.

Now that we have a plausible chemical mechanism, it is alarming to see how many unknowns there are. Basically, every primary molecular species requires one

parameter for the initial concentration. Every reaction step requires two parameters: k_f and k_b for binding and conversion reactions, and K_m and k_{cat} for enzyme reactions. If we represent enzymes as the explicit formation of an enzyme–substrate complex (as we highly recommend), then these two terms are insufficient; see equation (3.9). The approach we use is to assume a standard ratio between k_2 and k_3 , typically 4, unless there is direct experimental evidence on this rate.

How do we get all these parameters? If we are lucky, the numbers and indeed the whole model may exist in a database. It is curious that although the field of systems biology is relatively young, it has been far more successful than computational neuroscience in organizing data into such databases and in the use of community standards (De Schutter, 2008).

If no one has done the work of assembling parameters into models, then we have to resort to the library to look for experiments that define biochemical rates. Here again, we may get lucky and find that someone has measured the exact rates. Unfortunately, it is more likely that direct data are unavailable and we have to rely on indirect information, such as ratios of active to inactive forms of enzymes, or time courses of product formation.

There are a few common forms of biochemical data that work well for setting parameters.

1. Enzyme purification series. These experiments typically run tissue through a series of purification steps, ending with a fraction that is asymptotically pure. The final line in such a series will usually give values for the specific activity and the purification factor. These can be converted into V_{max} and concentration of the enzyme, respectively.
2. Activity ratios. Many signaling experiments are reported in terms of percentage change in activity (or concentration) following a stimulus. This will typically allow one to set a K_d or an enzyme activity term, depending on the underlying model.
3. Dose-response curves. Here the output (typically phosphorylation level) of the reaction is plotted against the concentration of an input molecule. If the underlying reaction model is simple (a binding step or two), then the half-saturation value of this curve gives a good estimate of the K_d of the rate-limiting step in the reaction. More complex dose-response curves are also useful constraints in a reaction scheme but cannot be used as directly.
4. Time series. Here the concentration of a molecule is plotted as a function of time, typically in the context of an input stimulus applied at some point. Another common data source that gives similar information is protein translocation data from microscopy studies. Such data provide many points for matching, which makes them good constraints for modeling. In the simplest case, this may look like an exponential set-

tling curve. From this one can obtain the time constant τ , which is $1/k_f$ if it is a simple binding reaction.

In most of the examples given here, a good reaction model will let one use the data even if the reaction scheme is somewhat complex. There now exists a wealth of parameter search techniques for fitting such data (chapter 2), but it is always preferable to use direct data when possible.

If we are truly unlucky, or on the bleeding edge, the reaction will not have been measured. This is the time to think about test tubes, or about collaborations with people who use them.

The usual caveats apply about all data sources, whether primary papers or databases:

- How similar were the experimental conditions (temperature, pH, reagents, isoforms, cell types, species)?
- Were the experiments done *in vitro*, *in tissue culture*, *in slice*, or *in vivo*?
- Do we have independent confirmation of the values?
- Do we trust the authors?

3.8 Conclusions

The brain is full of networks. We are used to the idea of networks of many neurons, and later chapters in this book discuss the characteristics and modeling methods applied to such systems. This chapter has introduced another whole set of networks: chemical circuits composed of interacting molecules. Each synapse in a neuron is home to a complex biochemical signaling network that is coupled to other synapses and to dendritic and somatic networks. Their study overlaps with many other fields of biology and other cell types, and the field is sometimes termed “systems biology”. The issues of levels of representation are particularly acute here, and in this chapter we have seen how this affects our choice of numerical tools. It is increasingly clear that the computations inside a cell offer as much richness, diversity, and challenges as those outside it.

4 Modeling Intracellular Calcium Dynamics

Erik De Schutter

In this chapter we use the methods developed in chapter 3 to model calcium dynamics inside neurons. A wide range of applications are covered, from simple phenomenological models used to drive the activation of calcium-gated channels to highly detailed modeling of events in a spine. Simulating calcium dynamics can be computationally costly and even for extensively studied cases, important parameters are not known (Neher, 1998); therefore it is quite important to choose the right level of complexity of model. We only consider the cytoplasmic calcium concentration C_i and keep the extracellular concentration C_o constant because changes of the latter are not thought to be relevant under *in vivo* conditions. We will first introduce how to model the different processes affecting the cytoplasmic calcium concentration and then consider a number of typical modeling cases.

4.1 Calcium Sources and Sinks

The free cytoplasmic calcium concentration C_i is the end result of the interaction of many processes, each of which needs to be described by its own set of equations in a complete model. Recent reviews of the biochemical properties of these processes can be found in Sabatini et al. (2001), Augustine et al. (2003), Berridge et al. (2003), and Hartmann and Konnerth (2005). Neuroscientists should realize that the literature on modeling calcium dynamics is vast and extends well beyond the neuroscience domain, with many interesting contributions from modeling the heart, muscles, and pancreatic beta cells.

Calcium Influx

Calcium can flow into the cytoplasm from two sources: through membrane channels or by release from internal stores. The second source is considered later.

Because of the huge calcium gradient across the cell membrane, calcium currents are strongly rectifying (Hille, 2001). Therefore one should always use the Goldman-

Hodgkin-Katz equation (GHK; equations 5.12–5.13 in chapter 5) to compute the current I_{Ca} . This will be more accurate than a linear ohmic current because, for example, the calcium current reversal predicted by the Nernst potential is quite unphysiological.

The change in a concentration that is due to ion entry through a membrane channel caused by an ion current I_{Ca} (chapter 5) is given by

$$\frac{dC_i}{dt} = -\frac{I_{\text{Ca}}(t)}{zFv}, \quad (4.1)$$

where z is the charge of the ion (+2 in the case of calcium), F is the Faraday constant (96,489 C · mol⁻¹), and v is the volume into which the flux occurs. An important parameter determining the rise and decay times of concentration changes caused by calcium currents is the surface-to-volume ratio (Cornelisse et al., 2007). The calcium influx scales with membrane surface as do the membrane pumps, but the effect of most removal systems like buffers and diffusion is related to the volume. Structures with a small volume (high surface-to-volume ratio) like spines and thin dendrites, will therefore, for the same density of calcium channels, show higher-amplitude changes that rise and decay faster than structures with high volumes, like a proximal dendrite.

For voltage-gated channels, usually the complete current is used to compute the calcium influx, but for many synaptic channels, the current is carried by a mixture of ions (chapter 6, section 1) and therefore the specific calcium component of the synaptic current is needed. This is most relevant for models that simulate calcium influx through NMDA and through some AMPA receptor channels (Burnashev, 1998) to induce synaptic plasticity (Holmes and Levy, 1990). The fractional calcium current, which expresses the relative charge carried by calcium at a given voltage, is in the range of 8–12% for NMDA channels (Burnashev, 1998; Sobczyk and Svoboda, 2007). This fraction is variable and can therefore not be directly applied to the synaptic current; this would also cause a reversal of the calcium current at the reversal potential of the synaptic channel (approximately 0 mV). Instead, one needs to compute the GHK equation for the calcium current through the synaptic channel and scale it by the time-varying synaptic conductance g_{syn} (Mayer and Westbrook, 1987; Badoual et al., 2006):

$$I_{\text{CaS}} = \bar{P}_{\text{Ca}} \frac{g_{\text{syn}}(t)}{\bar{g}_{\text{syn}}} G(V, Ca_o, Ca_i), \quad (4.2)$$

where \bar{P}_{Ca} is scaled to obtain the correct fraction of current at the reference voltage and $G(V, Ca_o, Ca_i)$ is the GHK current equation (equation 5.13).

Calcium Removal

The free cytoplasmic calcium at rest is very low (around 50 nM) because most of the calcium is bound to buffers (see later) or removed effectively by Na^+ - Ca^{2+} exchangers and ATP-driven pumps. Pumps are present on both the plasma membrane and the membrane of several organelles, with endoplasmic reticulum (Sabatini et al., 2002) and mitochondria (Friel, 2000) being the largest calcium sinks. Removal has a strong effect on calcium dynamics and in the case of small structures like spines it may even dominate the decay time of calcium signals (Sabatini et al., 2002). In general, neuron models have not been very sophisticated in simulating membrane transporters compared with the detailed modeling that is common for cardiac cells (e.g., DiFrancesco and Noble, 1985; and more recently the Kyoto model, which contains four membrane transporters and several leak currents; Takeuchi et al., 2006).

The Na^+ - Ca^{2+} exchanger (Blaustein and Lederer, 1999; Dipolo and Beaugé, 2006) is a low-affinity high-throughput system that uses the sodium gradient to remove calcium, with a stoichiometry of probably 3:1. The classic model by DiFrancesco and Noble (1985):

$$I_{\text{NaCa}} = k_{\text{NaCa}} ([\text{Ca}^{2+}]_o [\text{Na}^+]_i^3 e^{\gamma FV/RT} - [\text{Ca}^{2+}]_i [\text{Na}^+]_o^3 e^{(\gamma-1)FV/RT}) / (1 + 0.001([\text{Ca}^{2+}]_i [\text{Na}^+]_o^3 + [\text{Ca}^{2+}]_o [\text{Na}^+]_i^3)) \quad (4.3)$$

is still often used, with k_{NaCa} the maximum rate (in units of A mM⁻⁴, taken proportional to the membrane surface), []_o and []_i representing outside and inside concentrations, respectively, and the partition parameter γ representing the fractional position within the membrane of the voltage-sensitive energy barrier (Hille, 2001), usually taken to be 0.38. A disadvantage of this simple model is that it does not represent the affinities of the calcium and sodium binding sites, which are incorporated into more modern models (Cortassa et al., 2003; Takeuchi et al., 2006). Since the Na^+ - Ca^{2+} exchangers come in several molecular forms (Philipson and Nicoll, 2000), one can expect future models to be based on the kinetics of specific gene products. Usually only the exchange over the plasma membrane is modeled, but in reality exchange also occurs with internal organelles like mitochondria (Cortassa et al., 2003).

Equation (4.3) is formulated as a current and emphasizes, therefore, an important property of exchangers and pumps: they are electrogenic. In the case of the Na^+ - Ca^{2+} exchanger, this results in an inward current ($z = -1$) when $[\text{Ca}^{2+}]_i$ becomes high, although this depends on the exact stoichiometry of the exchange (Blaustein and Lederer, 1999).

The Ca-ATPase pumps are present both on the plasma membrane (PMCA family) and on the sarco/endoplasmic reticulum (SERCA family) (Strehler and Treiman, 2004). These high-affinity, low-capacity pumps are often modeled with a simple Hill (1910) equation:

$$\frac{dC}{dt} = V_{\max} \frac{C^n}{K_d + C^n}, \quad (4.4)$$

with V_{\max} as the maximum pump rate (units of moles per second taken proportional to the membrane surface), K_d the dissociation constant, and n the Hill constant (a measure of cooperativity between the binding sites). For $n = 1$, equation (4.4) reduces to the Michaelis-Menten function (chapter 3, section 2; Michaelis and Menten, 1913). Kinetic models for the different isoforms of SERCA pumps are becoming available (Dode et al., 2002; Yano et al., 2004); these models also include the inhibitory effect of the calcium store concentration on the pumps (Burdakov et al., 2005). Further elucidation of the structural basis of calcium movement by these pumps (Olesen et al., 2007) may lead to a new generation of kinetic models.

The PMCA pump generates an outward current ($z = 2$) and therefore it will counteract the current caused by the $\text{Na}^+ \text{-Ca}^{2+}$ exchanger. The net electrogenic effect of the pumps and exchangers is usually neglected in neural simulations but can nevertheless reach about 1% of the amplitude of I_{Ca} and modulate it significantly (De Schutter and Smolen, 1998).

One problem in using simple models of the exchanger and pumps, as in equations (4.3) and (4.4), is that they may drive calcium below its resting concentration. A common solution to this problem is to use a small leak from the endoplasmic reticulum, which is calibrated to give a stable resting concentration (Schmidt et al., 2003), but this may require rather large leak currents. Shannon et al. (2000) have proposed a more physiological model that includes the reverse mode of the SERCA pumps.

Calcium Release

Gated calcium release from internal stores in endoplasmic or sarcoplasmic reticulum can cause large changes in the internal calcium concentration. Calcium release is best known for its important role in the generation of calcium oscillations and calcium waves (Berridge et al., 2003; Fiacco and McCarthy, 2006), but it also contributes to the induction of synaptic plasticity in many systems (Bardo et al., 2006). Two important receptor channel types have been identified: inositol triphosphate (IP_3) (Taylor and Laude, 2002) and ryanodine (Fill and Copello, 2002) receptors, both of which have multiple subtypes. The expression of these receptors and their distribution are cell specific.

Calcium release is usually modeled as a ligand-dependent flux along the concentration gradient:

$$\frac{dC_i}{dt} = V_R f_o (C_s - C_i), \quad (4.5)$$

where V_R represents the maximum rate of release, f_o the fraction of open channels, and C_s the calcium concentration in the stores. Phenomenological models often use

a Hill function (equation 4.4) to relate f_o to the ligand concentration and equation (4.5) can then be solved as a simple ODE (chapter 1, section 6). For both receptor types, more complex Markov gating models (chapter 5, section 5) are available.

Another issue is how to model the calcium store concentration C_s . This concentration is quite high, 100–800 μM (Burdakov et al., 2005), and strongly buffered. The dynamics of calcium stores can influence the overall changes in cytoplasmic calcium concentration to a large degree; for example, in some models of calcium oscillations, the calcium release ends because of depletion of the store (Goldbeter et al., 1990). Recently it was confirmed that calcium release may indeed lead to highly localized decreases of the calcium store concentration (Brochet et al., 2005).

The IP_3 receptor is activated by both calcium and IP_3 . The source of the latter in neurons is activation of membrane-bound G-protein-linked receptors like the metabotropic glutamate receptors (mGluR) (Coutinho and Knöpfel, 2002). The calcium influx caused by IP_3 receptors has been modeled extensively to simulate calcium oscillations, first with phenomenological models (Goldbeter et al., 1990) and then with simple binding models (De Young and Keizer, 1992). The latter was the first model to include the dual action of calcium. At low concentrations it activates the IP_3 receptor, whereas higher calcium concentrations inactivate it (Taylor and Laude, 2002). Although this model is still extensively used, including a simplified version (Li and Rinzel, 1994), it has been outdated by the avalanche of recent molecular and biochemical data. Several more sophisticated models have been proposed recently (Doi et al., 2005) and have been reviewed by Sneyd and Falcke (2005). In addition to modeling the receptor, IP_3 metabolism needs to be simulated also. In many simulations this has been approximated as a pulse combined with a linear decay, but more detailed kinetic schemes have been proposed (Bhalla and Iyengar, 1999).

The molecular mechanisms underlying ryanodine receptor activation are less well known (Fill and Copello, 2002). It is suspected that ryanodine receptors are activated by a specific, unidentified signaling molecule, but they are best known as the receptors of calcium-induced calcium release (CICR) because their opening is strongly facilitated by increases of C_i and of C_s . Again, both phenomenological (Keizer and Levine, 1996) and complex gating models (Zahradník et al., 2005) have been proposed.

Calcium Buffering

Calcium buffering can be a source or a sink of calcium, depending on the balance between bound and free calcium. In fact, most of the cytoplasmic calcium is not free but is bound to different proteins. Neurons express a particular class of proteins called calcium buffers that are characterized by the presence of multiple EF-hand calcium-binding domains (Lewit-Bentley and Rety, 2000), for example, calbindin, calmodulin, and parvalbumin. The different calcium binding sites on these buffer

proteins will usually also have different rate constants and show cooperativity (Faas et al., 2007), but such complexity is often ignored (however, see W. R. Holmes, 2000; Schmidt et al., 2003; Naoki et al., 2005; Means et al., 2006; and Kubota et al., 2008 for models using binding site-specific rate constants). Under these simplifying conditions, the interaction of calcium with buffers is reduced to a second-order reversible reaction (chapter 3, section 2):



with a forward rate constant k_f and a backward rate constant k_b , which are related to the dissociation constant K_d of the buffer as $K_d = k_b/k_f$. An important issue is whether the buffer is mobile. Several experimentalists have the impression that cytoplasmic buffers are immobile because little or no washout is observed (Helmchen et al., 1996; Sabatini et al., 2002). In fact, however, most buffers have diffusion constants D_B in the range of 20–50 $\mu\text{m}^2/\text{s}$ (Schmidt et al., 2003), although the value for calmodulin is unclear (Naoki et al., 2005). Fractions of these buffers may be bound to cytoskeleton, but the majority of the molecules remain mobile (Schmidt et al., 2005). So for a complete model of calcium buffering, one requires for each buffer four parameters: the rate constants k_f and k_b , the diffusion constant D_B , and the total buffer concentration $[B]_T$ (the sum of $[B]$ and $[CaB]$). The first three parameters are known for the most common calcium buffers (e.g., Schmidt et al., 2003 for parvalbumin and calbindin and Means et al., 2006 and Tadross et al., 2008 for calmodulin or calreticulin) but $[B]_T$ is neuron specific and often a free parameter. A useful experimental value to constrain $[B]_T$ is the buffer capacity κ of the neuron, the fractional amount of bound over free calcium:

$$\kappa = \frac{d[CaB]}{d[Ca^{2+}]}. \quad (4.7)$$

Buffer capacity can be measured experimentally (Maravall et al., 2000) and ranges from 20 to 200 in most neurons (Neher, 1998; Sabatini et al., 2002) and up to 2,000 in Purkinje cells (Hartmann and Konnerth, 2005). Notice that κ may vary locally within neurons (Neher, 1998; Sabatini et al., 2002). Buffer capacity is related to the total buffer concentration for buffers with a low affinity (high K_d):

$$\kappa \approx \frac{[B]_T}{K_d} \quad \text{if } [Ca^{2+}] \ll K_d. \quad (4.8)$$

Calcium stores have high concentrations of specific buffers like calsequestrin and calreticulin, which have large numbers of calcium-binding sites but low affinity (Beard et al., 2004).

4.2 Calcium Diffusion

Diffusion of calcium, or of other signaling molecules like IP_3 , and of buffer molecules, can be simulated with any of the methods described in chapter 3, section 5. The diffusion constant D should be based on measurements in cytoplasm because this has a much higher viscosity than water; for example, D_{Ca} for cytoplasm is $2 \times 10^{-6} \text{ cm}^2 \text{ s}^{-1}$, which is three times slower than in water (Albritton et al., 1992). Note that the value for D_{Ca} used in the simulations may be very different from the experimentally observed apparent diffusion constant D_{app} , because many factors may affect the measured D_{app} (section 4.3 and Santamaria et al., 2006; Wagner and Keizer, 1994).

Numerical Solution and Dimensionality

In general, solving the diffusion equation is an expensive operation. We focus here only on deterministic solutions, but similar issues apply when stochastic methods are used (chapter 3, section 5). Numerical accuracy will require the use of smaller discretization steps both in the spatial (submicrometer) (Carnevale and Rosenthal, 1992) and time (about 1 μs) domains compared with the voltage equation (11.1), resulting in calculations that are several orders of magnitude slower (De Schutter and Smolen, 1998). For one- and two-dimensional diffusion, implicit methods can be used; explicit methods require even smaller time steps. The Crank-Nicholson method works quite well for the tridiagonal matrix describing a one-dimensional case (Fletcher, 1991; Press et al., 2007) and can be used in two dimensions with the alternating direction implicit (ADI) method (Press et al., 2007; see W. R. Holmes, 1995 for a neuroscience implementation). A specialized ADI implementation for modeling calcium dynamics in one to three dimensions is the program CalC (see the software appendix; Matveev et al., 2002). Deterministic modeling of three-dimensional diffusion usually employs finite-volume approaches based on slower, iterative solution methods (Ferziger and Peric, 2001).

Traditionally, many neural models have limited themselves to one-dimensional diffusion in modeling spines (W. R. Holmes and Levy, 1990; W. R. Holmes, 2000) or activation of membrane channels (De Schutter and Smolen, 1998; Yamada et al., 1998) because such models are relatively easy to solve. Of course this simplification does not allow accurate simulation of highly local phenomena like calcium nanodomains (Berridge, 2006; see section 3), which need a full 3-D simulation. Even in cases where only diffusion over long distances is of interest, ignoring some spatial dimensions may come at a risk. For example, a detailed 2-D simulation of IP_3 diffusion in Purkinje cell spiny dendrites (Hernjak et al., 2005) did not report the experimentally observed anomalous diffusion of IP_3 caused by trapping in spines (Santamaria et al., 2006). The discrepancy is explained by the difference in spine

density between 2-D and 3-D models of the spiny dendrite, which strongly influences the occurrence of anomalous diffusion (Santamaria et al., 2006). For simple geometric reasons, 2-D models can never approach the real spine density of Purkinje cells.

Electrodiffusion

Treating the computation of changes in membrane potential (chapter 10) and the diffusion of ions as separate processes is a simplification because the membrane electric field can exert an effect on charged molecules. Similarly, the axial current computed by the cable equation (10.2) is in reality caused by the drift of ions. Proper treatment of this interaction requires the use of the Nernst-Planck equation (Jack et al., 1975), which is rather cumbersome. Qian and Sejnowski (1989) have argued that in thin dendrites ($0.1\text{ }\mu\text{m}$) application of the full electrodiffusion equation is required to track fast changes in ionic concentration, and they proposed a modified cable equation. Most of the effects they describe, however, can already be simulated by the proper use of the Nernst or Goldman-Hodgkin-Katz equations (chapter 5, section 3) to track changes in the ion equilibrium potential.

Moreover, using the electrodiffusion equation forces the modeler to consider many cellular details that are normally ignored and that are often poorly characterized. Examples include the actual charge carriers in the cytoplasm, which may include small organic molecules like phosphates (De Schutter and Smolen, 1998; Lopreore et al., 2008), and the variable membrane potential of intracellular organelles (Shemer et al., 2008; Wikström et al., 1997; Yamashita et al., 2006). Since thermal velocities of molecules (causing the Brownian motion) are almost an order of magnitude larger than the drift velocities in an electrical field (Hille, 2001), it is unlikely that simulating the full electrodiffusion equation appreciably changes simulation results compared with the standard methods derived from the cable equation (Koch, 1999). Comparisons of a simulation of action potential generation at the node of Ranvier by the cable equation and by an electrodiffusion model (Lopreore et al., 2008) showed small differences in the width of the action potential, but at present it is not clear whether this prediction is physiologically relevant.

4.3 How to Choose the Right Approach to Modeling Calcium

It is difficult to model calcium accurately because most simulation approaches are really gross simplifications of reality. Calcium entry, whether through the plasma membrane or from internal sources, is a highly local process, occurring in nanodomains (below a single channel) and microdomains (below a cluster of channels; Augustine et al., 2003; Berridge, 2006). Influx from a single channel into the cytosol typically consists of about a thousand ions and causes a rise to a concentration of several hundred micromolars in a plume with dimensions of $\sim 10\text{ nm}$. This plume has been given

different names, depending on the source of calcium: sparks, puffs, sparklets, and scintillas (Berridge, 2006). Correct modeling of calcium concentrations in a nano- or microdomain remains a challenge because of the strong gradients, requiring finite-volume or stochastic approaches on very fine grids (Aharon et al., 1994; Meinrenken et al., 2002; Shahrezaei and Delaney, 2004).

Since some calcium sensors, such as Ca^{2+} -activated K^+ channels, sample concentration in these microdomains (Goldberg and Wilson, 2005; Zhuge et al., 2002; Fakler and Adelman, 2008), an accurate simulation would require a detailed, computationally expensive simulation. This is almost never feasible or desired and instead, modelers use simplified models of calcium dynamics. The question then becomes: What is the proper simplification to use?

Simple Calcium Pools to Simulate Channel Activation by Calcium

Compartmental neuron models (chapter 11, section 1) often need to compute calcium concentration to activate Ca^{2+} -dependent channels (chapter 5) in the plasma membrane. Since the calcium concentration itself is usually not of primary interest, the model can be highly simplified. The most commonly used model is the exponentially decaying Ca^{2+} pool (Traub and Llinás, 1977):

$$\frac{dC_i}{dt} = -\frac{I_{\text{Ca}}(t)}{2Fv} - \beta(C_i - C_{\min}). \quad (4.9)$$

This model contains a biophysical component, the influx (compare with equation 4.1), and a phenomenological one, the removal of calcium. The influx is scaled by a volume v . Although it may seem that v is fixed by the size of the electrical compartment, it is in practice a free parameter because for large compartments it makes more sense to consider the calcium concentration only in the submembrane region, where it is sampled by the Ca^{2+} -activated channels. Therefore v is often taken as the volume of a submembrane shell about 0.2 μm deep (De Schutter and Bower, 1994a), thinner shells causing steeper changes in calcium concentration. The removal of calcium is governed by the parameter β , which lumps together the effects of buffers, pumps, and diffusion. β is the inverse of the time constant of exponential decay and therefore this parameter will determine how fast calcium returns to the resting concentration C_{\min} . A wide range of values are used for this parameter; decay time constants ($1/\beta$) range from 0.1 ms (De Schutter and Bower, 1994a) to 50 ms (Traub and Llinás, 1977). Fast decay time constants combined with thin submembrane shells make the computed calcium concentration track calcium influx closely, which is a reasonable approximation of the changes in microdomain concentration (Sherman et al., 1990) and can be used to activate BK-type Ca^{2+} -activated potassium channels (Zhuge et al., 2002; Fakler and Adelman, 2008).

One problem with the calcium pool is that it reduces the calcium dynamics to a single time constant. If, for example, multiple types of Ca^{2+} -activated K^+ channels need to be simulated, this may cause a problem because these typically show different dynamics. SK-type Ca^{2+} -activated potassium channels activate slower than BK channels and the difference is attributed to the indirect activation by calcium (Fakler and Adelman, 2008). A simple and effective solution is to use two or more calcium pools with different values for β (Tegnér and Grillner, 1999). This can be implemented in different ways. One can distribute the calcium influx over the pools or one can attribute specific pools to different sources of calcium (Tegnér and Grillner, 1999). Although this approach is attractive, it increases the number of purely phenomenological parameters in the model. Therefore, especially if the data are available, one should instead consider a more detailed calcium buffering model in which the different affinities of the Ca^{2+} -activated channels make them sense varying time constants of the system (see the following discussion).

Calcium Buffering in a Single Compartment

If the binding properties of the intrinsic calcium buffers are known, as is now the case for an increasing number of neurons (Cornelisse et al., 2007; Schmidt et al., 2003), it is more realistic to model a system with calcium buffers and pumps at only a little added computational cost (figure 4.1). Each additional buffer introduces two new

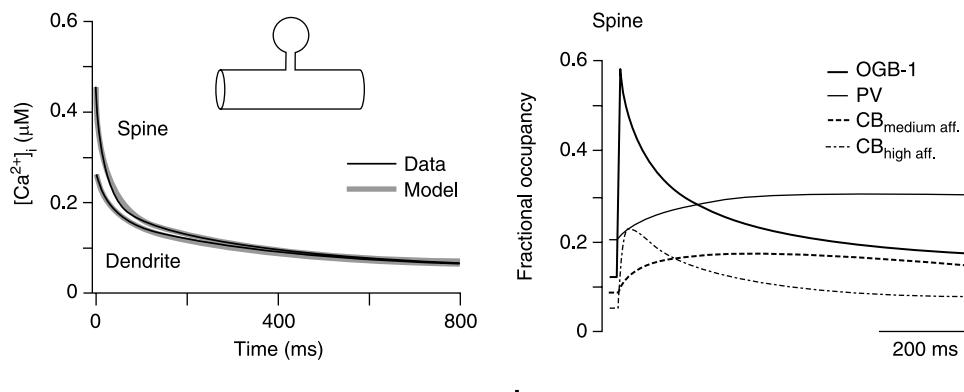


Figure 4.1

Simulation of calcium decay kinetics in a Purkinje cell spine. The model is parametrized based on comparing calcium imaging data in normal and transgenic mice and includes three diffusible buffers: the calcium dye (OGB-1), calbindin (CB, two binding sites), and parvalbumin (PV). Panel a shows that the model replicates well the biphasic calcium decay observed experimentally in both spine and dendrite. Panel b shows the calcium binding of the different buffers in the spine. It indicates that the biphasic decay is caused by the different binding kinetics of the two intrinsic buffers and that the measurements in spines suffer from saturation of the calcium dye. (Modified with permission from Schmidt et al. 2003.)

time constants to the system, causing more complex calcium dynamics in the model. The forward rate of the buffer will be critical in determining the maximum rise in calcium concentration. An exception is buffer saturation, i.e., when most of the buffer calcium binding sites become occupied, because then the system reverts to the unbuffered behavior (Bormann et al., 2001).

Removal of calcium can be modeled as a single high-rate calcium pump. Such a single-compartment system ignores the effects of diffusion, but this simplification is justified in some structures, such as a dendritic spine (Sabatini et al., 2002). If one adds a few more molecular features to such a simulation, e.g., calcium release from intracellular stores, it becomes a valuable tool in a systems biology approach to simulate signaling pathways involved in the induction of synaptic plasticity (Doi et al., 2005).

For larger structures where calcium influx across the cell membrane is the primary source, diffusion toward the center becomes more important (Yamada et al., 1998). In principle, one could approximate the removal that is due to diffusion phenomenologically (equation 4.9) and use this single-compartment buffering approach to model the calcium concentrations needed to activate potassium channels, but this approach has not yet been tested in large compartmental models (chapter 11, section 1).

Buffered Calcium 1-D Diffusion to Model Calcium Imaging Data

A more realistic approach is to include 1-D diffusion of calcium and the mobile buffers into the model, but this steeply increases the computational cost. Such systems have been investigated extensively, using both analytical (Naraghi and Neher, 1997; Wagner and Keizer, 1994; Zador and Koch, 1994) and modeling (Nowycky and Pinter, 1993) approaches. An important concept introduced by these studies is the apparent diffusion constant for buffered calcium diffusion, D_{app} :

$$D_{app} = \frac{1}{1 + \kappa} D_{Ca} + \kappa D_B, \quad (4.10)$$

where the buffer capacity κ was defined in equations (4.7) and (4.8). D_{app} is a convenient way to describe how binding to buffers affects calcium diffusion. However, the simplification is valid only for the steady state in the presence of unsaturated buffers with a fast forward rate (Wagner and Keizer, 1994) and therefore in most models a full simulation is warranted (Bormann et al., 2001). In the presence of an immobile buffer ($D_B = 0$), equation (4.10) predicts a slowing down of the calcium diffusion because of calcium binding to the buffer. This effect steeply limits the spatial extent to which calcium can affect chemical reactions in the neuron compared with other signaling molecules with similar diffusion constants that are not buffered, such as IP₃ (Kasai and Petersen, 1994; Santamaria et al., 2006). It is interesting that calcium

seems to diffuse effectively over long distances, although slowly, in the endoplasmic reticulum (Choi et al., 2006; Mogami et al., 1997) despite its assumed high buffer capacity.

The mobility of buffers ($D_B > 0$ in equation 4.10) can reverse this situation because they may increase D_{app} by shuttling calcium from high concentration regions (where they bind calcium) to low concentration regions (where they unbind it) (Bormann et al., 2001; Wagner and Keizer, 1994).

This is one of the problems that calcium dyes introduce. Most calcium dyes act as mobile buffers and may modify the calcium dynamics being observed by changing both buffer capacity and apparent diffusion (A. Takahashi et al., 1999). One solution is to use low-affinity dyes only, but this excludes measuring small changes in concentration from that at rest (S. S. Wang et al., 2000). An alternative is to use modeling methods to estimate the error (H. Blumenfeld et al., 1992; Cornelisse et al., 2007; Sala and Hernandez-Cruz, 1990). This can even be done with a spreadsheet (McHugh and Kenyon, 2004; <http://www.medicine.nevada.edu/physio/docs/diffusion.htm>). The basis of these models is 1-D buffered calcium diffusion in a spherical cell (Yamada et al., 1998), combined with a transfer function to compute the fluorescence from the concentrations of bound and unbound calcium dye (Grynkiewicz et al., 1985) and, possibly, a model of the filtering properties of the imaging equipment (H. Blumenfeld et al., 1992).

Buffered 1-D diffusion has also been used to simulate calcium transients in dendritic spines, where the spine is modeled using an FD approach (chapter 3, section 5) by stacking a set of thin cylinders on each other (W. R. Holmes and Levy, 1990; Zador et al., 1990). Although such simulations have provided useful insights, they are being replaced with more detailed 3-D models (Keller et al., 2008).

A Full 3-D Model to Simulate Calcium Signaling Pathways

This is the ne plus ultra in calcium modeling and the appropriate tool for modeling the effect of calcium nano- and microdomains. It is, however, not only challenging to compute these complex models, but also to design them. The number of molecular species simulated and the specification of parameters and boundary conditions make the development and tuning of these models as demanding as large compartmental models (chapter 11).

As a result, many recent examples simulate only a small part of a neuron, i.e., signaling events in a spine (Ichikawa, 2005; Keller et al., 2008) or calcium entry in a presynaptic terminal (Bennett et al., 2000; Meinrenken et al., 2002). It is interesting that stochastic simulation is being used extensively in neuroscience for such models, with MCell or STEPS (see the software appendix). This is quite different from the systems biology approaches, which focus more on the use of finite-element methods

in unstructured meshes to attempt to model signaling in large parts of reconstructed cells and organelles (Hernjak et al., 2005; Means et al., 2006).

Based on continuous improvements in calcium imaging techniques, combined with the increase in available computer speed and the rapid development of new software tools for 3-D simulation, we expect to see exciting new, detailed calcium models in the near future.

5 Modeling Voltage-Dependent Channels

Alain Destexhe and John R. Huguenard

5.1 Introduction

Many different types of voltage-dependent ion channels have been identified and are responsible for a rich repertoire of electrical behavior essential for neuronal function (Llinás, 1988). Modeling voltage-dependent ion channels is crucial for assessing their numerous roles in the genesis of the complex intrinsic properties of central neurons, as well as how such neurons integrate synaptic inputs with intrinsic excitability to generate spike output. The seminal work of Hodgkin, Huxley, Katz, and others several decades ago still constitutes the basis of today's models. The first accurate model of membrane excitability was introduced by Hodgkin and Huxley (1952) and was based on relatively simple biophysical mechanisms underlying the Na^+ and K^+ conductances that generate action potentials in the giant axon of the squid. This model reproduced well the behavior of the recorded currents, and its parameters are easy to determine from experimental data. This explains why Hodgkin-Huxley models are still widely used today, almost sixty years later.

The postulate of the Hodgkin-Huxley model was that membrane currents result from the assembly of gating particles freely moving in the membrane. The molecular components responsible for ionic permeabilities were later identified as being transmembrane protein complexes containing a pore specifically permeable to one or several ions (reviewed in Hille, 2001). These ion channels can have their permeability modulated by various factors, such as the voltage or the binding of a ligand. The sensitivity of some ion channels to voltage is a fundamental property that constitutes the core mechanism underlying the electrical excitability of membranes and is still today an important matter of investigation (for a review, see Armstrong and Hille, 1998). In particular, the Hodgkin-Huxley model was cast in a formalism more compatible with statistical physics and thermodynamics (Tsien and Noble, 1969; T. L. Hill and Chen, 1972; Stevens, 1978), which we call here thermodynamic models.

Single-channel recording techniques (reviewed in Sakmann and Neher, 1995) provided significant advances in our understanding of the biophysical properties of ion

channels. Single-channel recordings have shown that ion channels display rapid and stochastic transitions between conducting and nonconducting states. It is now known that conformational changes of the channel protein give rise to opening and closing of the channel. Conformational changes in ion channels can be described by state diagrams analogous to the conformational changes underlying the action of enzymes (chapter 3, section 2). Markov models are based on such transition diagrams and have been used for modeling various types of ionic currents based on single-channel recordings (for a complete overview, see Sakmann and Neher, 1995). This formalism is more accurate than Hodgkin-Huxley models, but its drawback is the greater difficulty of estimating its parameters from experimental data (Cannon and D'Alessandro, 2006). However, Markov models can also be used to draw simplified representations of the current, which capture only the most salient properties of voltage-dependent or synaptic interactions (Destexhe et al., 1994). Such simplified models are more adequate for representing currents when simulating networks involving thousands of cells.

Thus, various formalisms of different levels of complexity have been proposed to model ionic currents. Which formalism to adopt for modeling a given current depends on the experimental data available and its accuracy, as well as the desired level of precision in the behavior of the model. We illustrate these aspects in this chapter by considering different types of formalisms in modeling such processes as the action potential and voltage-clamp recordings of the T-type calcium current in thalamic neurons. For both cases, we show the similarities and differences among the different models, how well they account for experimental data, and which is the “minimal” model needed to reproduce electrophysiological behavior.

5.2 The Hodgkin-Huxley Formalism

The Hodgkin-Huxley Model

In a remarkable series of experiments on the squid giant axon, Hodgkin, Huxley, and colleagues determined that ionic conductances can be activated or inactivated according to the membrane potential. They used the technique of a voltage clamp to record the ionic currents generated at different voltages and thus infer how these currents can be dynamically modulated by voltage. They characterized the kinetics of two voltage-dependent currents, the fast sodium current, I_{Na} , and the delayed potassium rectifier, I_K , mediated by Na^+ and K^+ ions, respectively. A mathematical model was necessary to establish that the identified kinetic properties of voltage dependence were sufficient to explain the genesis of action potentials. The model introduced by Hodgkin and Huxley (1952) incorporated the results of their voltage-clamp

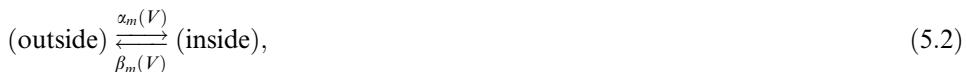
experiments and successfully accounted for the main properties of action potentials, which represented very convincing evidence that their postulated mechanism was plausible.

The Hodgkin-Huxley model is based on a membrane equation describing three ionic currents in an isopotential compartment:

$$c_m \frac{dV}{dt} = -g_L(V - E_L) - g_{Na}(V)(V - E_{Na}) - g_K(V)(V - E_K), \quad (5.1)$$

where c_m is the membrane capacitance; V is the membrane potential; g_L , g_{Na} , and g_K are the membrane conductances for leak currents, Na^+ , and K^+ currents respectively; and E_L , E_{Na} , and E_K are their respective reversal potentials, which are given by the Nernst relation (equation 5.11).

The critical step in the Hodgkin-Huxley model is to specify how the conductances $g_{Na}(V)$ and $g_K(V)$ depend on the membrane potential V . Hodgkin and Huxley hypothesized that ionic currents result from the assembly of several independent gating particles that must occupy a given position in the membrane to allow the flow of Na^+ or K^+ ions (Hodgkin and Huxley, 1952). Each gating particle can be on either side of the membrane and bears a net electronic charge such that the membrane potential can switch its position from the inside to the outside or vice versa. The transition from these two states is therefore voltage dependent, according to the diagram:



where α and β are, respectively, the forward and backward rate constants for the transitions from the outside to the inside position in the membrane. If m is defined as the fraction of particles in the inside position, and $(1 - m)$ as the fraction outside, one obtains the first-order kinetic equation:

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m. \quad (5.3)$$

Assuming that particles must occupy the inside position to conduct ions, then the conductance must be proportional to some function of m . In the case of the squid giant axon, Hodgkin and Huxley (1952) found that the nonlinear behavior of the Na^+ and K^+ currents, their delayed activation, and their sigmoidal rising phase were best fit by assuming that the conductance is proportional to the product of several of such variables:

$$g_{Na} = \bar{g}_{Na}m^3h \quad (5.4)$$

$$g_K = \bar{g}_K n^4, \quad (5.5)$$

where \bar{g}_{Na} and \bar{g}_K are the maximal values of the conductances, while m , h , and n represent the fraction of three different types of gating particles in the inside of the membrane. This equation allowed them to accurately fit the voltage-clamp data of the currents. Their interpretation is that the assembly of three gating particles of type m and one of type h is required for Na^+ ions to flow through the membrane, while the assembly of four gating particles of type n is necessary for the flow of K^+ ions. These particles operate independently of each other, leading to the m^3h , and n^4 forms.

Long after the work of Hodgkin and Huxley, when it was established that ionic currents are mediated by the opening and closing of ion channels, the gating particles were reinterpreted as *gates* inside the pore of the channel. Thus, the reinterpretation of Hodgkin and Huxley's hypothesis was that the pore of the channel is controlled by four internal gates, that these gates operate independently of each other, and that all four gates must be open in order for the channel to conduct ions.

The rate constants $\alpha(V)$ and $\beta(V)$ of m and n are such that depolarization promotes opening the gate, a process called *activation*. On the other hand, the rate constants of h are such that depolarization promotes closing of the gate (and therefore closing of the entire channel because all gates must be open for the channel to conduct ions), a process called *inactivation*. Thus the experiments of Hodgkin and Huxley established that three identical activation gates (m^3) and a single inactivation gate (h) are sufficient to explain the Na^+ current's characteristics. The squid axon K^+ current does not have inactivation and can be well described by four identical activation gates (n^4).

Taking together all the steps here, one can write the following set of differential equations, called Hodgkin-Huxley equations (Hodgkin and Huxley, 1952):

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - E_L) - \bar{g}_{Na} m^3 h (V - E_{Na}) - \bar{g}_K n^4 (V - E_K) \\ \frac{dm}{dt} &= \alpha_m(V)(1-m) - \beta_m(V)m \\ \frac{dh}{dt} &= \alpha_h(V)(1-h) - \beta_h(V)h \\ \frac{dn}{dt} &= \alpha_n(V)(1-n) - \beta_n(V)n. \end{aligned} \quad (5.6)$$

Hodgkin and Huxley (1952) estimated the rate constants (α_i and β_i) by fitting empirical functions of voltage to the experimental data. These functions are

$$\begin{aligned}
 \alpha_m &= \frac{-0.1(V - V_r - 25)}{\exp[-(V - V_r - 25)/4] - 1} \\
 \beta_m &= 4 \exp[-(V - V_r)/18] \\
 \alpha_h &= 0.07 \exp[-(V - V_r)/20] \\
 \beta_h &= \frac{1}{1 + \exp[-(V - V_r + 30)/10]} \\
 \alpha_n &= \frac{-0.01(V - V_r + 10)}{\exp[-(V - V_r + 10)/10] - 1} \\
 \beta_n &= 0.125 \exp[-(V - V_r)/80].
 \end{aligned} \tag{5.7}$$

These functions were estimated at a temperature of 6°C (see section 5.7 for temperature dependence) and the voltage axis has been reversed in polarity (voltage values were given with respect to the resting membrane potential, V_r) compared to the original study.

The Hodgkin-Huxley model is often written in a form more convenient to fit to experimental data by rewriting equation (5.3) in the equivalent form:

$$\frac{dm}{dt} = \frac{1}{\tau_m(V)} [m_\infty(V) - m], \tag{5.8}$$

where

$$m_\infty(V) = \alpha(V)/[\alpha(V) + \beta(V)] \tag{5.9}$$

$$\tau_m(V) = 1/[\alpha(V) + \beta(V)]. \tag{5.10}$$

Here, m_∞ is the *steady-state activation* and τ_m is the *activation time constant* of the Na^+ current (n_∞ and τ_n represent the same quantities for the K^+ current). In the case of h , h_∞ and τ_h are called steady-state inactivation and inactivation time constant, respectively. These quantities are important because they can easily be determined from voltage-clamp experiments. The Boltzmann equation (Hille, 2001; equation 2.1) is commonly used for $m_\infty(V)$.

Fitting to Voltage-Clamp Data

We now turn to the problem of how to use the results of voltage-clamp experiments to build a Hodgkin-Huxley type of model. We discuss these protocols and illustrate them for another type of current, the low-threshold calcium current, also called the T-type calcium current, T-current, or I_T . To this end, we use data collected

previously (Huguenard and Prince, 1992) using whole-cell patch recordings from acutely dissociated thalamic relay neurons from the ventrobasal thalamus of young rats (P8–P15). All recordings correspond to a temperature of 24 °C (see Huguenard and Prince, 1992 for details about those experiments).

Activation

Like Hodgkin-Huxley's Na^+ current, the T-current is transient and activates upon depolarization, but is slower and its voltage range for activation and inactivation typically occurs around resting membrane potential. To reveal activation properties, the typical voltage-clamp protocol is to clamp the membrane using a series of voltage steps from a hyperpolarized level (-100 mV) to various depolarized levels. Such a protocol reveals an inward current that activates and inactivates in a voltage-dependent manner (figure 5.1a2); i.e., each process becomes faster with stronger depolarizations.

Deactivation

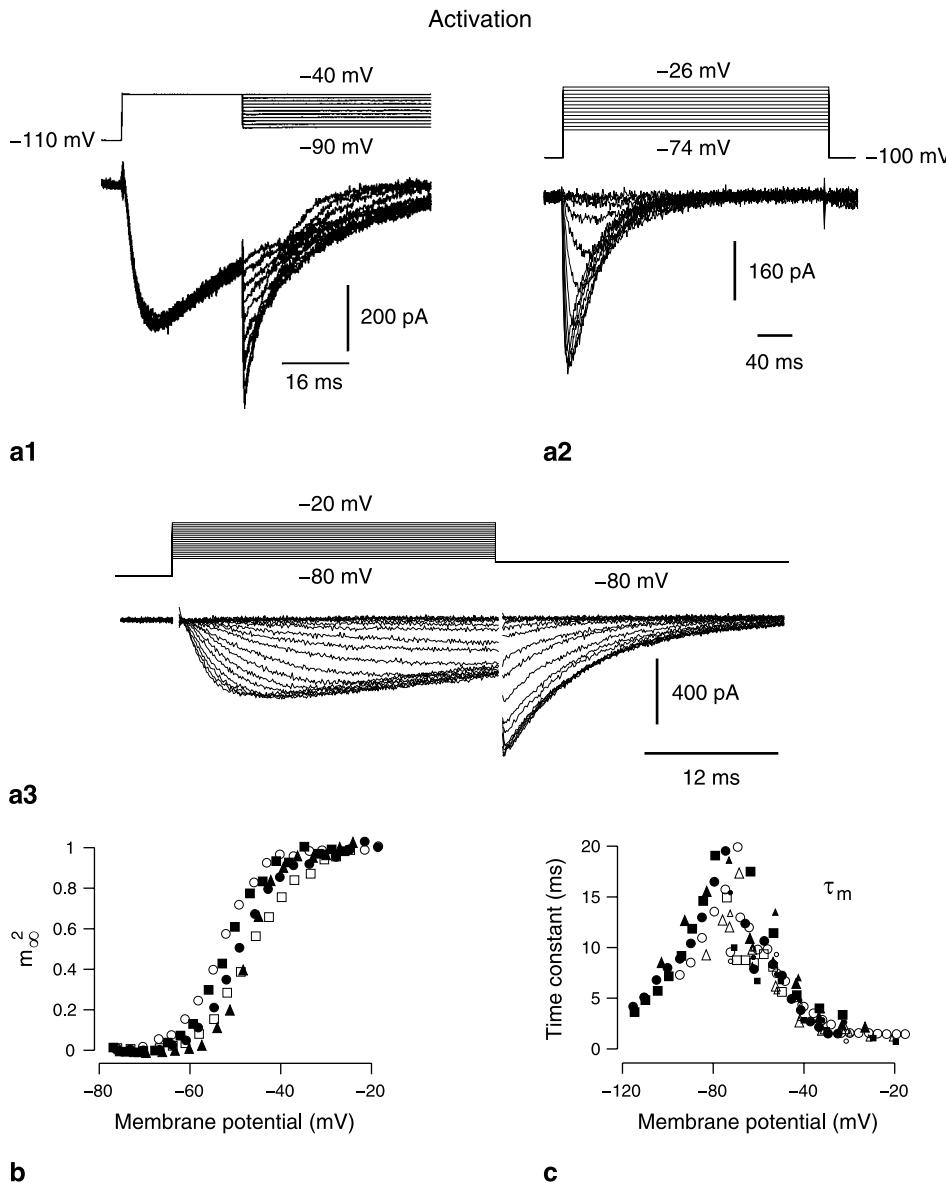
Interrupting the activation protocol before inactivation is complete generates tail currents (figure 5.1a1) that reveal the deactivation characteristics of the current. Deactivation is the reversal of the activation process: the m gating particles revert to a closed state during membrane hyperpolarization. Because inactivation is relatively slow (see next), the kinetics of tail currents, which reflect channel closing, at hyperpolarized membrane potentials are predominately dependent on deactivation, which is much faster. Deactivation time constants obtained from tail current analysis are plotted on the left side (i.e., between membrane potentials of -120 to -80 mV) of the bell-shaped curve in figure 5.1c.

Inactivation

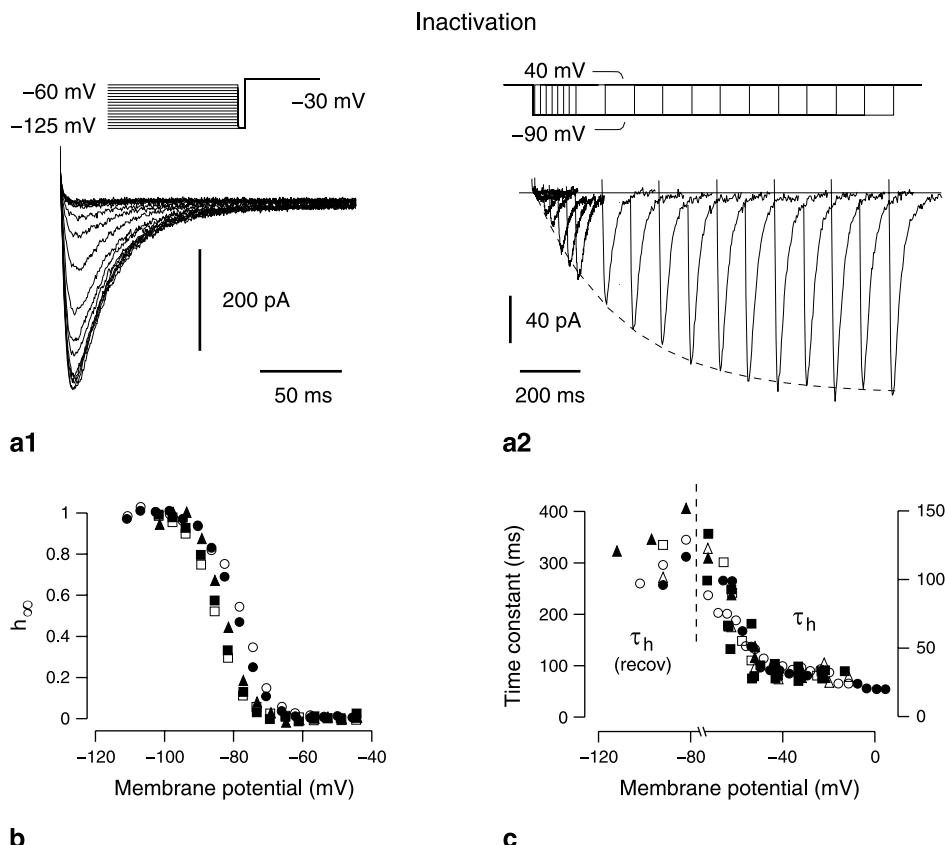
The typical voltage-clamp protocol to reveal steady-state inactivation is to apply a series of holding potentials for a prolonged period (several seconds) to allow inactivation to obtain steady-state equilibrium. Then a command potential is applied that activates the current (at -30 mV in this case). This protocol is shown in figure 5.2a1 for the T-current. The different current traces obtained contain similar activation but different levels of inactivation.

Deinactivation

By analogy to deactivation, deinactivation represents the reversal of the inactivation process: h particles revert to an open state during membrane hyperpolarization. The kinetics of deinactivation can be measured by reactivating the current following complete inactivation. This is accomplished by first holding the membrane potential at a depolarized level (-40 mV here) to inactivate the current, followed by step voltage-clamp commands to a hyperpolarized value (-90 mV) for variable amounts of time

**Figure 5.1**

Voltage-clamp recordings of the T-current in dissociated thalamic relay neurons. (a) Voltage-clamp protocols for determining deactivation (a1), voltage-dependent activation and inactivation rates (a2), and steady-state activation (a3). Command potentials at various levels were given after the cell was maintained at a hyperpolarized holding potential, leading to the activation of the current. (b) Steady-state activation obtained from the tail currents in a3, which were fit to an m^2h template. (c) Time constants obtained using a similar procedure. Different symbols correspond to different cells. (Figure modified from Huguenard and McCormick, 1992, where all details are given.)

**Figure 5.2**

Voltage-clamp characterization of T-current inactivation in dissociated thalamic relay neurons. (a) Voltage-clamp protocols for inactivation (a1) and recovery from inactivation (a2). In A1, the cell was maintained at different holding potentials then stepped to -30 mV to activate the T-current with different levels of inactivation. In a2, the current was reactivated after being fully inactivated. The full recovery took about 1 s (recovery time constant of about 300 ms). (b) Steady-state inactivation calculated by the peak of currents in a1. (c) Inactivation time constants obtained by fitting an $m^2 h$ template to the data. The recovery time constants were obtained by fitting a single exponential to the recovery experiment (dashed line in a2). Different symbols correspond to different cells. (Figure modified from Huguenard and McCormick, 1992, where all details are given.)

before returning to the initial depolarized level. The second depolarization generates an inward current whose amplitude is proportional to the amount of recovery from inactivation. A particular feature of the T-current is that this recovery from inactivation is very slow (hundreds of millisecond; figure 5.2a2).

Estimating Steady-State Values and Time Constants

Steady-state values and time constants were estimated as follows. First, we assumed that T-current gating could be approximated by the Hodgkin-Huxley formalism for sodium channels, with multiple activation gates and a single inactivation gate (equation 5.4). The optimal number of activation gates was determined by examining differences in residuals between the original current traces and curves best-fitted to various templates (i.e., m^2h , m^3h , and m^4h). This approach originally suggested that m^3h provided the best fit to the data (Coulter et al., 1989). However, later results obtained under conditions that better isolated the T-current (Huguenard and Prince, 1992) proved that the m^2h scheme was more appropriate. Thus the optimal template for the T-current included two activation gates and one inactivation gate (Huguenard and Prince, 1992).

Next, to measure activation, the influence of inactivation must be as minimal as possible. We assumed that activation is essentially complete in 10 ms, and that there is negligible inactivation (these assumptions were checked by calculating the expected activation and inactivation at various voltages). We used the amplitude of the tail current, which reflects the number of channels open at the end of the depolarizing step, as a measure of activation (m^2). The values obtained using this procedure were very close to those obtained by direct fitting of Hodgkin-Huxley equations to current traces (Huguenard and Prince, 1992). The advantage of the tail current approach is that the driving force is the same for all measurements, therefore providing a direct measure of normalized conductance. This type of procedure leads to estimates of steady-state activation (figure 5.1b).

Steady-state inactivation was obtained by plotting the peak current amplitude (obtained at a test potential of -30 mV) as a function of conditioning potential in figure 5.2b). The current traces from which these values were obtained are shown in figure 5.2a1).

The time constants (figures 5.1c and 5.2c) were estimated by fitting the full expression of the current (from equation 5.12) to the current traces (figure 5.1a2), while allowing m and h to evolve as a function of time according to equation (5.3) (see the methods section in Huguenard and McCormick, 1992). Once the steady-state and time constant values are obtained, one must fit either the empirical functions of voltage as done by Hodgkin and Huxley (1952) or by using predefined templates as predicted by theoretical arguments (see section 5.7).

5.3 Implementation

Nernst Equation

In the original Hodgkin-Huxley model, it was assumed that the currents are ohmic, i.e., they vary linearly as a function of voltage and conductance. In such a case, the reversal potential is given by the Nernst equation. For example, for K⁺ ions:

$$E_K = \frac{RT}{zF} \ln \frac{[K]_o}{[K]_i}, \quad (5.11)$$

where $R = 8.31 \text{ J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$ is the gas constant, T is the absolute temperature in kelvins, z is the valence of the ion ($z = 1$ for K⁺ ions, $z = -1$ for Cl⁻ ions, etc.), $F = 96,489 \text{ C} \cdot \text{mol}^{-1}$ is the Faraday constant, and $[K]_o$ and $[K]_i$ are the concentrations of K⁺ ions outside and inside the membrane, respectively (see chapter 3).

Goldman-Hodgkin-Katz Equation

While many ionic currents are ohmic, some show clear deviations from ohmic behavior, which appears as a deviation from linearity in the I–V representation of the current. This deviation, also called *rectification*, may be due to diverse factors, such as very high ionic concentration gradients across the membrane. This is the case for calcium currents in central neurons; the internal and external Ca²⁺ concentrations differ by about four orders of magnitude, and as the ion channels open or close, the membrane never reaches equilibrium (which would be given by the Nernst equation). In such far-from-equilibrium situations, one must use a different formalism. The simplest of such nonlinear models is the *constant-field equation*, also called Goldman-Hodgkin-Katz equation (see details in Hille, 2001). The current is given by

$$I_T = \bar{P}_{\text{Ca}} m^2 h G(V, Ca_o, Ca_i), \quad (5.12)$$

where \bar{P}_{Ca} (in centimeters per second) is the maximum permeability of the membrane to Ca²⁺ ions (the permeability here is $\bar{P}_{\text{Ca}} m^2 h$, the product of maximum permeability and the fraction of channels in an open state), and $G(V, Ca_o, Ca_i)$ is a nonlinear function of voltage and ionic concentrations:

$$G(V, Ca_o, Ca_i) = z^2 F^2 V / RT \frac{Ca_i - Ca_o \exp(-zFV/RT)}{1 - \exp(-zFV/RT)}, \quad (5.13)$$

where $z = 2$ is the valence of calcium ions. Ca_i and Ca_o are the intracellular and extracellular Ca²⁺ concentrations (in molar concentrations), respectively.

5.4 Thermodynamic Models

In the Hodgkin-Huxley model, the rate constants $\alpha(V)$ and $\beta(V)$ were fit to the experimental data by using empirical functions of voltage. An alternative approach is to deduce the exact functional form of the voltage dependence of the rate constants from thermodynamics. These thermodynamic models (Tsien and Noble, 1969; T. L. Hill and Chen, 1972; Stevens, 1978) provide a plausible physical basis for constraining and parameterizing the voltage dependence of rate constants, which are then used to fit voltage-clamp experiments.

In thermodynamic models, the transition between two states of the channel corresponds to a conformational change in the ion channel protein. Consider a transition between an initial (I) and a final (F) state, with a rate constant $r(V)$ that is voltage dependent:



According to the theory of reaction rates (Eyring, 1935; F. H. Johnson et al., 1974), the rate of the transition depends exponentially on the free-energy barrier between the two states:

$$r(V) = r_0 e^{-\Delta G(V)/RT}, \quad (5.15)$$

where r_0 is a constant and $\Delta G(V)$ is the free-energy barrier, which can be written as

$$\Delta G(V) = G^*(V) - G_0(V), \quad (5.16)$$

where $G^*(V)$ is the free energy of an intermediate state (activated complex), and $G_0(V)$ is the free energy of the initial state, as illustrated in figure 5.3. The relative values of the free energy of the initial and final states (G_0 and G_1) determine the equilibrium distribution between these states, but the kinetics of the transition depend on the size of the free-energy barrier $\Delta G(V)$. Systems with a smaller energy barrier (figure 5.3, dashed line) correspond to faster kinetics because a larger proportion of molecules will have the energy required to form the activated complex and make the transition.

In ion channels, these different states correspond to different conformations of the ion channel protein. How the transition rates between these conformational states depend on membrane potential is given by the voltage dependence of the free-energy barrier, which is in general difficult to evaluate. The effect of the electrical field on a protein will depend on the number and position of its charged amino acids, which will result in both linear and nonlinear components in the free energy. Without assumptions about the underlying molecular structure, the free energy of a given state i can be written as a Taylor series expansion of the form:

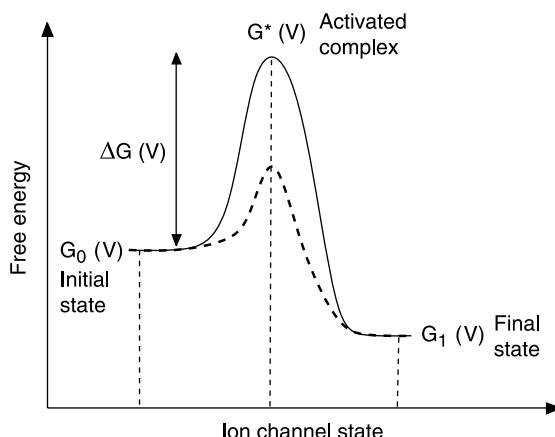


Figure 5.3

Schematic representation of the free-energy profile of conformational changes in ion channels. The diagram represents the free energy of different states involved in a transition: the initial state, activated complex, and final state. The equilibrium distribution between initial and final states depends on the relative value of their free energy (G_0 and G_1). The rate of the transition will be governed by the free-energy barrier ΔG , which is the free-energy difference between the activated complex and the initial state. If the energy barrier is smaller (dashed line), the kinetics of the reaction is faster because a larger proportion of ion channels will have the energy required to make the transition. (Figure modified from Destexhe and Huguenard, 2000.)

$$G_i(V) = A_i + B_i V + C_i V^2 + \dots \quad (5.17)$$

where A_i, B_i, C_i, \dots are constants that are specific for each conformational state. The constant A_i corresponds to the free energy that is independent of the electrical field; the linear term $B_i V$ corresponds to the interaction between an electrical field with isolated charges and rigid dipoles (Tsien and Noble, 1969; T. L. Hill and Chen, 1972; Stevens, 1978; Andersen and Koeppe, 1992). For example, linear terms in V will result if the conformations differ in their net number of charges, or if the conformational change is accompanied by the translation of a freely moving charge inside the structure of the channel (T. L. Hill and Chen, 1972; Hille, 2001). Nonlinear terms result from such effects as electronic polarization and pressure induced by V (T. L. Hill and Chen, 1972; Stevens, 1978; Andersen and Koeppe, 1992) or from mechanical constraints associated with the movement of charges that are due to the structure of the ion channel protein (Destexhe and Huguenard, 2000).

Thus, each conformational state of the ion channel protein will be associated with a given distribution of charges and will therefore be characterized by a given set of coefficients in equation (5.17). This is also true for the activated state, which is a particular case of conformation. Applying equations (5.15)–(5.17), the rate constant becomes

$$\begin{aligned} r(V) &= r_0 e^{-[(A^* + B^* V + C^* V^2 + \dots) - (A_0 + B_0 V + C_0 V^2 + \dots)]/RT} \\ &= r_0 e^{-(a+bV+cV^2+\dots)/RT}, \end{aligned} \quad (5.18)$$

where $a = A^* - A_0, b = B^* - B_0, c = C^* - C_0, \dots$ represent differences between the linear and nonlinear components of the free energy of the initial and activated states (according to equation 5.17).

Consider the particular case of a reversible open-closed transition



where C and O are, respectively, the closed and open states, and α and β are the forward and backward rate constants. Applying equation (5.18) to forward and backward reactions leads to the following general expression for the voltage dependence:

$$\begin{aligned} \alpha(V) &= \alpha_0 e^{-(a_1+b_1V+c_1V^2+\dots)/RT} \\ \beta(V) &= \beta_0 e^{-(a_2+b_2V+c_2V^2+\dots)/RT}, \end{aligned} \quad (5.20)$$

where $a_1, a_2, b_1, b_2, c_1, c_2, \dots$ are constants specific to this transition. It is important to note that in general these parameters are not necessarily interrelated because the three different conformations implicated here (initial, activated, and final, as in figure 5.3) may have very different distributions of charges, resulting in different coefficients in equation (5.17), and thus also resulting in different values for $a_1 \dots c_2$. In the following discussion, this general functional form for the voltage dependence of rate constants is called the *nonlinear thermodynamic model* (see Destexhe and Huguenard, 2000).

In the “low field limit” (during relatively small transmembrane voltages), the contribution of the higher-order terms may be negligible. Thus, a simple, commonly used voltage dependence results from the first-order approximation of equation (5.20) and takes the form:

$$\begin{aligned} \alpha(V) &= \alpha_0 e^{-(a_1+b_1V)/RT} \\ \beta(V) &= \beta_0 e^{-(a_2+b_2V)/RT}. \end{aligned} \quad (5.21)$$

In the following discussion, this form with a simple exponential voltage dependence of the rate constants will be called the *linear thermodynamic model*.

A further simplification is to consider that the conformational change consists of the movement of a gating particle with a charge q (Hodgkin and Huxley, 1952; see also Borg-Graham, 1991). The forward and backward rate constants then become

$$\begin{aligned}\alpha(V) &= \alpha_0 e^{-\gamma qFV/RT} \\ \beta(V) &= \beta_0 e^{(1-\gamma)qFV/RT},\end{aligned}\tag{5.22}$$

where γ is the relative position of the energy barrier in the membrane (between zero and one). The constants α_0 and β_0 can be equated to a fixed constant A by introducing the half-activation voltage V_H , leading to

$$\begin{aligned}\alpha(V) &= Ae^{-\gamma qF(V-V_H)/RT} \\ \beta(V) &= Ae^{(1-\gamma)qF(V-V_H)/RT}.\end{aligned}\tag{5.23}$$

This form was introduced by Borg-Graham (1991) for modeling the gating of ion channels. Its parameters are convenient for fitting experimental data: V_H and q affect the steady-state activation and inactivation curves whereas A and γ only affect the time constant, with no effect on steady-state relations.

The drawback of models in which the rate functions are simple exponentials of voltage is that these functions can reach unrealistically high values, which leads to very small time constants and possibly aberrant behavior. A possible way to solve this problem is to force an artificial saturation of the rate constants (Willms et al., 1999) or impose a minimum value on the time constant (Borg-Graham, 1991).

Another possibility, physically more plausible, is not to limit the approximation of equation (5.20) to linear terms, but include higher-order terms in the voltage dependence of the free energy (Destexhe and Huguenard, 2000). For example, the quadratic expansion of equation (5.20) can be written as

$$\begin{aligned}\alpha(V) &= Ae^{-[b_1(V-V_H)+c_1(V-V_H)^2]/RT} \\ \beta(V) &= Ae^{[b_2(V-V_H)+c_2(V-V_H)^2]/RT},\end{aligned}\tag{5.24}$$

and similarly, its cubic expansion:

$$\begin{aligned}\alpha(V) &= Ae^{-[b_1(V-V_H)+c_1(V-V_H)^2+d_1(V-V_H)^3]/RT} \\ \beta(V) &= Ae^{[b_2(V-V_H)+c_2(V-V_H)^2+d_2(V-V_H)^3]/RT},\end{aligned}\tag{5.25}$$

where $A, b_1 \dots d_2$ are constants as defined earlier.

In addition to the effect of voltage on isolated charges or dipoles, described in equation (5.21), these forms account for more sophisticated effects, such as the deformation of the protein by the electrical field (T. L. Hill and Chen, 1972; Stevens, 1978) or mechanical constraints on charge movement (Destexhe and Huguenard, 2000). It also makes it possible for the model to capture more complicated depen-

dence on voltage than the simple exponential functions of equation (5.21), which may result in more realistic behavior (see section 5.7).

Finally, another way to impose a minimal value on the time constant is to consider that the gate operates via two successive transitions:



where C_1 and C_2 are two distinct closed states of the gate. The second transition is not dependent on voltage and therefore acts as a rate-limiting factor when α and β are large compared with k_1 and k_2 . In this case, the system will be governed essentially by k_1 and k_2 , which therefore impose a limit on the rate of opening and closing of the gate. On the other hand, when α and β are small compared with k_1 and k_2 , the system will be dominated by the first transition, while the two states C_2 and O will be in rapid quasi-equilibrium. Although this system apparently solves the problem of having a minimal time constant while still conserving the voltage dependence of the gate, it is nevertheless still unrealistic that the simple exponential representation for α and β permits the first transition to occur arbitrarily fast at some voltages.

Reaction schemes involving multiple states, such as equation (5.26), are reminiscent of another class of models, called Markov models, which are described in more detail in the next section.

5.5 Markov Models

As outlined earlier, the formalism introduced by Hodgkin and Huxley (1952) was remarkably forward looking and closely reproduced the behavior of macroscopic currents. However, Hodgkin-Huxley models are not exact and in fact rest on several approximations, and some of their features are inconsistent with experiments. Measurements on Na^+ channels have shown that activation and inactivation must necessarily be coupled (Armstrong, 1981; Aldrich et al., 1983; Bezanilla, 1985), which is in contrast to the independence of these processes in the Hodgkin-Huxley model. Na^+ channels may also show an inactivation that is not voltage dependent, as in the Hodgkin-Huxley model, but state dependent (Aldrich et al., 1983). Although the latter can be modeled with modified Hodgkin-Huxley kinetics (Marom and Abbott, 1994), these phenomena are best described using Markov models, a formalism more appropriate for describing single channels.

Markov models represent the gating of a channel as occurring through a series of conformational changes of the ion channel protein and assume that the transition probability between conformational states depends only on the present state. The

sequence of conformations involved in this process can be described by state diagrams of the form:

$$S_1 \rightleftharpoons S_2 \rightleftharpoons \cdots \rightleftharpoons S_n, \quad (5.27)$$

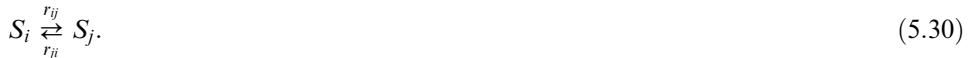
where $S_1 \dots S_n$ represents distinct conformational states of the ion channel. Defining $P(S_i, t)$ as the probability of being in a state S_i at time t and $P(S_i \rightarrow S_j)$ as the transition probability from state S_i to state S_j ($j = 1 \dots n$), according to



leads to the following equation for the time evolution of $P(S_i, t)$:

$$\frac{dP(S_i, t)}{dt} = \sum_{j=1}^n P(S_j, t) P(S_j \rightarrow S_i) - \sum_{j=1}^n P(S_i, t) P(S_i \rightarrow S_j). \quad (5.29)$$

This equation is called the *master equation* (see e.g., Stevens, 1978; Colquhoun and Hawkes, 1981). The left term represents the “source” contribution of all transitions entering state S_i , and the right term represents the “sink” contribution of all transitions leaving state S_i . In this equation, the time evolution depends only on the present state of the system and is defined entirely by knowledge of the set of transition probabilities (Markovian system). In the limit of large numbers of identical channels, the quantities given in the master equation can be replaced by their macroscopic interpretation. The probability of being in a state S_i becomes the fraction of channels in state S_i , noted s_i , and the transition probabilities from state S_i to state S_j become the rate constants, r_{ij} , of the reactions



In this case, one can rewrite the master equation as

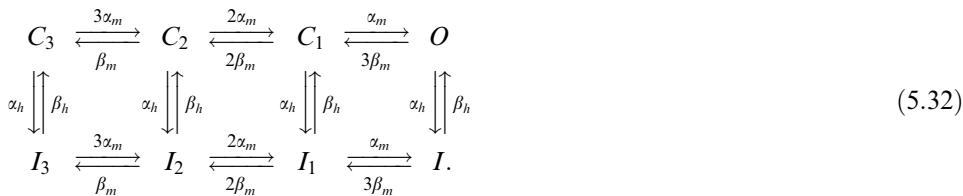
$$\frac{ds_i}{dt} = \sum_{j=1}^n s_j r_{ji} - \sum_{j=1}^n s_i r_{ij}, \quad (5.31)$$

which is a conventional kinetic equation for the various states of the system. Note that the rate constants can be voltage dependent and can be expressed as earlier (e.g., equation 5.20).

Stochastic Markov models (as in equation 5.29) are adequate to describe the stochastic behavior of ion channels as recorded using single-channel recording techniques (see Sakmann and Neher, 1995). In other cases, where a larger area of

membrane is recorded and large numbers of ion channels are involved, the macroscopic currents are nearly continuous and are more adequately described by conventional kinetic equations, as in equation (5.31) (see Johnston and Wu, 1995). In the following discussion, only systems of the latter type will be considered.

Note that Markov models are more general than the Hodgkin-Huxley formalism and include it as a subclass. Any Hodgkin-Huxley model can be written as a Markov scheme (while the opposite is not true). For example, the Markov model corresponding to the Hodgkin-Huxley sodium channel is (Fitzhugh, 1965):



Here, the different states represent the channel with the inactivation gate in the open state (top) or closed state (bottom) and (from left to right) three, two, one, or none of the activation gates closed. To be equivalent to the m^3 formulation, the rates must have the 3:2:1 ratio in the forward direction and the 1:2:3 ratio in the backward direction. Only the O state is conducting.

The squid delayed rectifier potassium current modeled by Hodgkin and Huxley with four activation gates and no inactivation can be treated analogously (Fitzhugh, 1965; Armstrong 1969), giving

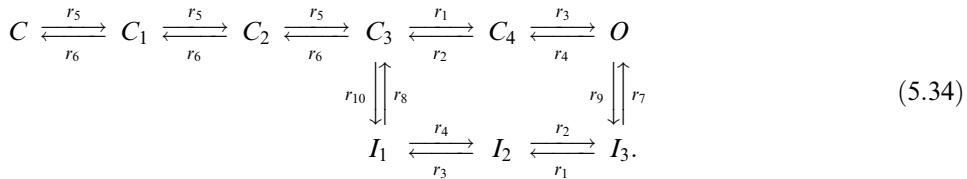
$$C_4 \xrightleftharpoons[3\beta_m]{4\alpha_m} C_3 \xrightleftharpoons[2\beta_m]{3\alpha_m} C_2 \xrightleftharpoons[3\beta_m]{2\alpha_m} C_1 \xrightleftharpoons[4\beta_m]{\alpha_m} O. \quad (5.33)$$

5.6 Fitting Models to Experimental Data

We now use some of the formalisms reviewed here and compare them in similar situations. Two situations are considered successively. The first is the voltage-clamp behavior of the sodium channel and the genesis of action potentials, and the second consists of the characteristics of the T-type calcium current and the genesis of bursts of action potentials by the T-current in thalamic neurons.

Models of Na^+ and K^+ Currents Underlying Action Potentials

We compare here the Hodgkin-Huxley model with two Markov models of Na^+ channels. A nine-state Markov model was proposed by Vandenberg and Bezanilla (1991):



This particular nine-state model was selected to fit not only the measurements of macroscopic ionic currents available to Hodgkin and Huxley, but also recordings of single-channel events and measurements of currents resulting directly from the movement of charge during conformational changes of the protein (so-called *gating currents*; see Hille, 2001). The voltage dependence of the transition rates was assumed to be a simple exponential function of voltage (equation 5.21).

To complement the sodium channel model of Vandenberg and Bezanilla, we also examined the six-state scheme for the squid delayed rectifier channel used by Perozo and Bezanilla (1990):



where again rates were described by a simple exponential function of voltage (equation 5.21).

The third class of model considered here consists of simplified Markov models of Na^+ and K^+ currents. The model for the Na^+ channel was chosen to have the fewest possible number of states (three) and transitions (four) while still being capable of reproducing the essential behavior of the more complex models. The form of the state diagram was based on a looped three-state scheme in which some transitions were eliminated, giving an irreversible loop (Bush and Sejnowski, 1991; Destexhe et al., 1994):



This model incorporated voltage-dependent opening, closing, and recovery from inactivation, while inactivation was voltage independent. For simplicity, neither opening from the inactivated state nor inactivation from the closed state was permitted. Although there is clear evidence for occurrence of the latter (Horn et al., 1981), it was unnecessary under the conditions of the present simulations. Rate constants were described by

$$r_i(V) = \frac{a_i}{1 + \exp[-(V - c_i)/b]}, \quad (5.37)$$

with $c_1 = c_2$ to yield a model consisting of nine total parameters (Destexhe et al., 1994).

The simplified K⁺ channel model consisted in a single open or conducting state O , and a single closed state C :



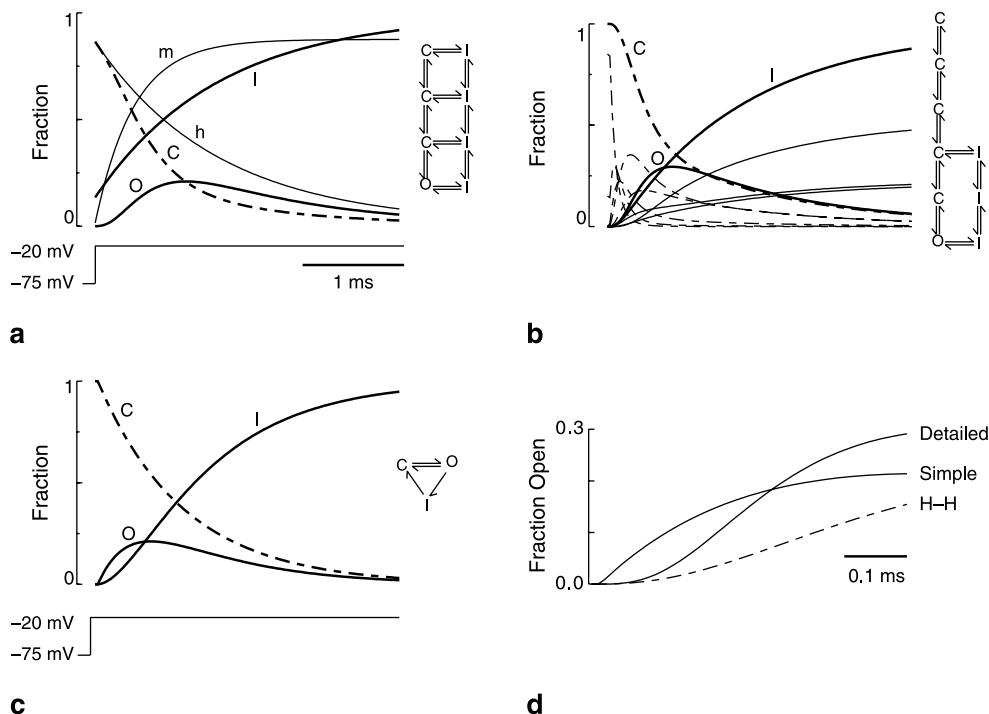
Here, the rates $r_1(V)$ and $r_2(V)$ had a sigmoidal voltage dependence similar to that in equation (5.37) (see details in Destexhe et al., 1994).

Na⁺ Currents in a Voltage Clamp

The different types of models reviewed here are characterized by different complexity, ranging from a two-state representation (equation 5.38) to transition diagrams involving many states (equation 5.34). The two-state description is adequate for the behavior of some channels (see Labarca et al., 1985; Yamada et al., 1998; Borg-Graham, 1991; Destexhe et al., 1994; Destexhe et al., 1998a), but for most channels more complex models must be considered. To illustrate this, we compared three different models of the fast sodium channel underlying action potentials (figures 5.4 and 5.5).

The responses of the three sodium channel models were compared during a voltage-clamp step from resting potential (-75 mV) to a depolarized level of -20 mV (figure 5.4). For all three models, the closed states were favored at hyperpolarized potentials. Upon depolarization, forward (opening) rates sharply increased whereas closing (backward) rates decreased, causing a migration of channels in the forward direction toward the open state. The three closed states in the Hodgkin-Huxley model and the five closed states in the Vandenberg-Bezanilla model gave rise to the characteristic delayed activation and sigmoidal shape of the rising phase of the sodium current (figure 5.4d). In contrast, the simple model, with a single closed state, produced a first-order exponential response to the voltage step and was therefore not sigmoidal.

These models generate different predictions about single-channel behavior. The steady-state behavior of the Hodgkin-Huxley model of the macroscopic sodium current is remarkably similar to that of the Vandenberg-Bezanilla (1991) model, but there are important differences in the relationship between activation and inactivation. First, as mentioned earlier, in the Hodgkin-Huxley model, activation and inactivation are kinetically independent. This independence has been shown to be untenable on the basis of gating and ion current measurements in the squid giant

**Figure 5.4**

Three kinetic models of a squid axon sodium channel produce qualitatively similar conductance time courses. A voltage-clamp step from rest, $V = -75 \text{ mV}$, to $V = -20 \text{ mV}$ was simulated. The fraction of channels in the open state (O , thick solid line), closed states (C , thick dashed lines), and inactivated states (I , thick dotted lines) are shown for the Hodgkin-Huxley model (1952), a detailed Markov model (Vandenberg and Bezanilla, 1991), and a simple Markov model (Destexhe et al., 1994). (a) Hodgkin-Huxley model of the sodium channel (equation 5.32). The activation (m) and inactivation (h) gates were deduced from other states and are indicated by thin lines. (b) Markov model of Vandenberg and Bezanilla (1991; equation 5.34). Individual closed and inactivated states are shown (thin lines), as well as the sum of all five closed states (C), the sum of all three inactivated states (I) and the open state (O). (c) Simplified three-state Markov model (Destexhe et al., 1994; equation 5.36). (d) Comparison of the time course of open channels for the three models on a faster time scale shows differences immediately following the voltage step. The Hodgkin-Huxley (H-H) and Vandenberg-Bezanilla (detailed) models give smooth, multiexponentially rising phases, while the three-state Markov model (simple) gives a single exponential rise with a discontinuity in the slope at the beginning of the pulse. (Figure modified from Destexhe et al., 1994, where all details are given.)

axon (Armstrong, 1981; Aldrich et al., 1983; Bezanilla, 1985). Consequently, Markov models that reproduce gating currents, such as the Vandenberg-Bezanilla model examined here, require schemes with coupled activation and inactivation. Likewise, in the simple model, activation and inactivation were strongly coupled, owing to the unidirectional looped scheme (equation 5.36), so that channels were required to open before inactivating and could not reopen from the inactivated state before closing.

A second difference is that in the Hodgkin-Huxley and Vandenberg-Bezanilla models, inactivation rates are slow and activation rates are fast. In the simplified Markov model, the situation was reversed, with fast inactivation and slow activation. At the macroscopic level modeled here, these two relationships gave rise to similar time courses for open channels (figure 5.4a–c; see Andersen and Koeppe, 1992). However, the two classes of models make distinct predictions for single-channel behavior. Whereas the Hodgkin-Huxley and Vandenberg-Bezanilla models predict the latency to the first channel opening to be short and channel open times to be comparable to the time course of the macroscopic current, the simplified Markov model predicts a large portion of first channel openings to occur after the peak of the macroscopic current and to have open times much shorter than the current's duration.

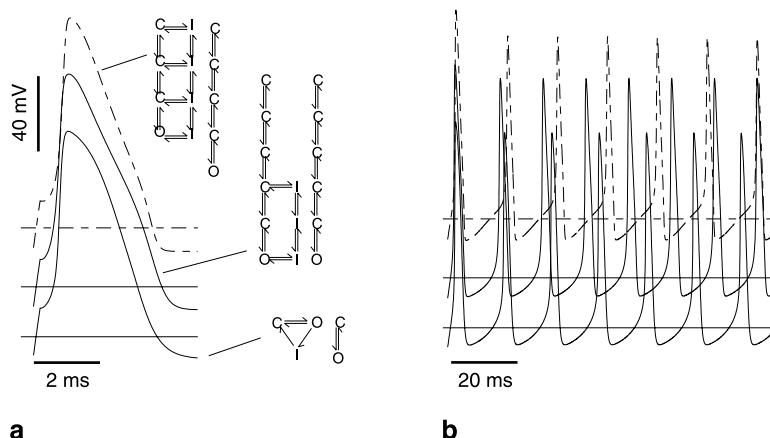
Genesis of Action Potentials (Current Clamp)

Despite significant differences in their complexity and formulation, the three models of the sodium channel all produced comparable action potentials and repetitive firing when combined with appropriate delayed-rectifier potassium channel models (figure 5.5). These simulations thus seem to perform similarly for fitting the macroscopic behavior of Na^+ and K^+ currents.

However, these three models generated clear differences when compared in a voltage clamp (figure 5.4) and still larger differences would be expected at the single-channel level. Thus, the choice of model clearly depends on the scope of the modeling study. If the detailed behavior of voltage-clamp experiments or single-channel recordings is to be reproduced, Markov models are certainly the most appropriate representation. However, if the goal is to reproduce the qualitative features of membrane excitability, action potentials, and repetitive firing, all models seem equivalent, except that simpler models are faster to compute. Thus in this case, simplified two- or three-state schemes or the Hodgkin-Huxley model would seem most appropriate.

5.7 Models of the T-Type Calcium Current

The different formalisms reviewed earlier are now applied to the example of voltage-clamp experiments of the T-type (low threshold) calcium current responsible for bursting behavior in thalamic neurons (Jahnsen and Llinás, 1984).

**Figure 5.5**

Similar action potentials produced using three different kinetic models of squid fast sodium and delayed rectifying potassium channels. (a) Single action potentials in response to 0.2-ms, 2-nA current pulse are elicited at similar thresholds and produce similar waveforms using three different pairs of kinetic models: Hodgkin and Huxley (1952) (dot-dashed line), detailed Markov models (Perozo and Bezanilla, 1990; Vandenberg and Bezanilla, 1991) (dotted line), and simplified kinetic models (solid line). (b) Repetitive trains of action potentials elicited in response to sustained current injection (0.2 nA) have slightly different frequencies. Sodium channels were modeled as described in figure 5.4. The detailed Markov potassium channel model had six states (Perozo and Bezanilla, 1990; equation 5.35) and the simple model of potassium channel had two states (equation 5.38). (Figure modified from Destexhe et al., 1994, where all details are given.)

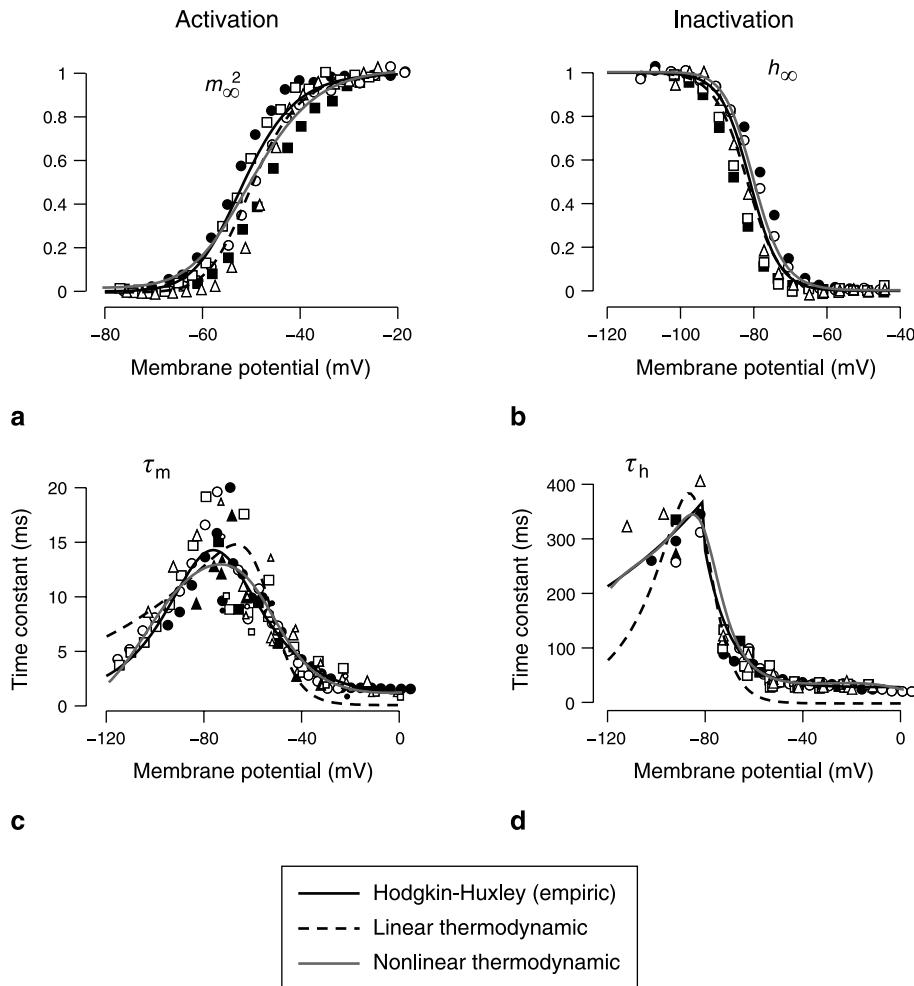
Hodgkin-Huxley Model of the T-Current

The voltage-clamp behavior shown here was first modeled by a Hodgkin-Huxley type of representation in which rate constants were fit to experimental data using empirical functions of voltage (Huguenard and McCormick, 1992). Owing to the non-linear behavior of calcium currents (the internal and external Ca^{2+} concentrations differ by about four orders of magnitude), they were represented using the constant-field equation (equation 5.12). The variables m and h in this equation represent, respectively, the activation and inactivation variables and obey first-order equations similar to equation (5.8). Their steady-state relations were fit using Boltzmann equations (figure 5.6a–b, thin solid lines), leading to the following optimal functions:

$$m_\infty(V) = 1/(1 + \exp[-(V + 57)/6.2]) \quad (5.39)$$

$$h_\infty(V) = 1/(1 + \exp[(V + 81)/4]). \quad (5.40)$$

Similarly, the voltage-dependent time constants were estimated by fitting exponential functions to the values determined experimentally (figure 5.6c–d, thin solid lines), leading to the following expression for activation:

**Figure 5.6**

Fitting of different models to the T-current in thalamic relay neurons. In each panel, the symbols show the voltage-clamp data obtained in several thalamic neurons (see figures 5.1 and 5.2) and the continuous curves show the best fits obtained with an empirical Hodgkin-Huxley-type model (thin solid line), a linear thermodynamic model (dashed line), and a nonlinear thermodynamic model (thick solid line). (a) Steady-state activation (m_∞^2). (b) Steady-state inactivation (h_∞). (c) Activation time constant (τ_m). (d) Inactivation time constant (τ_h). The leftmost symbols in d (≤ -80 mV) are the data from the slow recovery from inactivation of the T-current. See the text for the values of the parameters. All functions were fit using a simplex method. (Figure modified from Destexhe and Huguenard, 2000.)

$$\tau_m(V) = 0.612 + 1/(\exp[-(V + 132)/16.7] + \exp[(V + 16.8)/18.2]) \quad (5.41)$$

and for inactivation:

$$\begin{aligned} \tau_h(V) &= 28 + \exp[-(V + 22)/10.5] \quad \text{for } V \geq -81 \text{ mV} \\ &\quad \exp[(V + 467)/66.6] \quad \text{for } V < -81 \text{ mV}. \end{aligned} \quad (5.42)$$

Here two different functions were fit to the time constants τ_h obtained from inactivation protocols ($V \geq -81$) or recovery from inactivation ($V < -81$).

The temperature dependence of these empirical functions was adjusted according to the following rule:

$$\tau' = \tau Q_{10}^{-(T-24)/10}, \quad (5.43)$$

where Q_{10} is the experimentally determined change of time constants for a 10-degree difference in temperature. For the T-current in thalamic neurons, Q_{10} was determined as equal to 5 for τ_m and 3 for τ_h (Coulter et al., 1989).

The behavior of this model is shown in figure 5.7a. The model accounted well for all protocols of figures 5.1 and 5.2, with activation and recovery from inactivation shown in figures 5.7a1 and a2, respectively. However, in this model, τ_m and τ_h were fit using functions of voltage obtained empirically. Similar to the work of Hodgkin and Huxley (1952), this approach leads to a model that accounts well for the current-clamp behavior of the T-current in thalamic neurons (McCormick and Huguenard, 1992; see figure 5.8).

Linear Thermodynamic Model of the T-Current

Another possibility is to deduce the functional form of rate constants from thermodynamic arguments. The first of such models is the linear approximation. Constraining the fitting using rate constants described by equation (5.21) (figure 5.6, dashed lines) led to the following optimal expressions:

$$\alpha_m = 0.049 \exp[444\gamma_m(V + 54.6)/RT] \quad (5.44)$$

$$\beta_m = 0.049 \exp[-444(1 - \gamma_m)(V + 54.6)/RT] \quad (5.45)$$

$$\alpha_h = 0.00148 \exp[-559\gamma_h(V + 81.9)/RT] \quad (5.46)$$

$$\beta_h = 0.00148 \exp[559(1 - \gamma_h)(V + 81.9)/RT], \quad (5.47)$$

where $\gamma_m = 0.90$ and $\gamma_h = 0.25$. The steady-state relations and time constants are obtained similarly to those in equations (5.9–5.10).

This model provided a good fit of the steady-state relations (figure 5.6a–b, dashed lines), but the fit to time constants was poor (figure 5.6c–d, dashed lines). In particular,

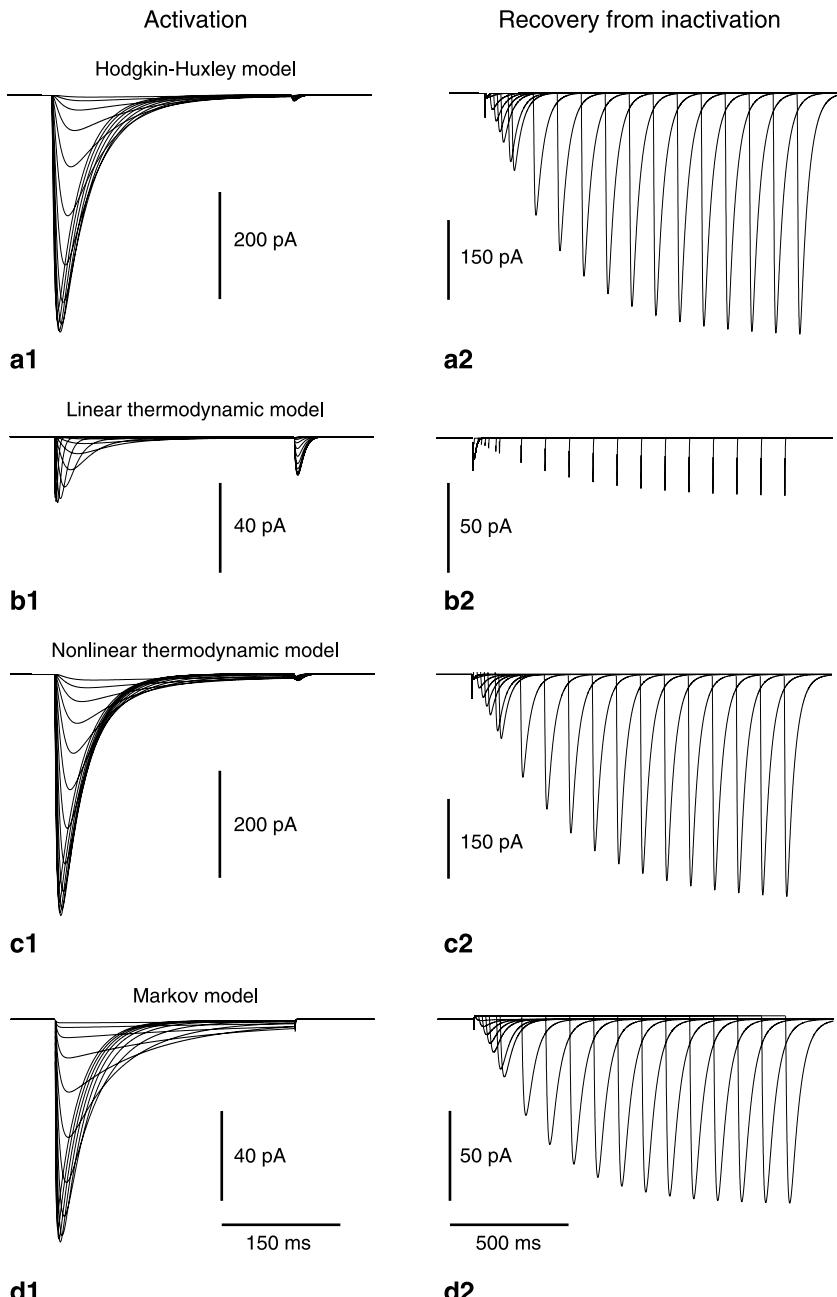


Figure 5.7

Voltage-clamp behavior of different models of the T-current. Left panels: activation protocol (identical to figure 5.1a2); right panels: protocol for the recovery from inactivation (identical to figure 5.2a2). (a) Empirical Hodgkin-Huxley-type model. (b) Linear thermodynamic model. (c) Nonlinear thermodynamic model. (d) Markov model. In all cases, the same density of T-channels was used ($\bar{P}_{\text{Ca}} = 3 \times 10^{-6} \text{ cm/s}$). (Figure modified from Destexhe and Huguenard, 2000.)

it was not possible to capture the saturation of τ_m and τ_h to constant values for depolarized membrane potentials. This poor fit had catastrophic consequences, as illustrated in figure 5.7b. Owing to the near-zero time constants at depolarized levels, the current activated and inactivated too fast and led to peak current amplitudes that were more than an order of magnitude smaller than the Hodgkin-Huxley model at the same channel densities (compare a and b in figure 5.7). We conclude that linear thermodynamic models do not provide an acceptable behavior in a voltage clamp for the T-current.

One possibility of resolving this inherent limitation is to add an artificial minimum value to the time constant (Borg-Graham, 1991), but this possibility was not considered here in order to stay within a physically plausible formalism. Instead, we show next that this problem can be solved by including higher-order voltage-dependent contributions in the free-energy barrier (Destexhe and Huguenard, 2000).

Nonlinear Thermodynamic Model of the T-Current

Nonlinear thermodynamic models assume that the free-energy barrier depends nonlinearly on voltage (see equation 5.20) and that each conformational state involved has its own dependence on voltage, independently of other conformational states (Destexhe and Huguenard, 2000). The consequence is that the coefficients $a_1 \dots c_2$ in equation (5.20) can take any value independently of each other. Using these nonlinear expressions to fit the voltage-clamp data of the T-current led to better fits of T-channel data. The quadratic expansion still provided a poor fit of the time constants, although better than linear fits (not shown). Acceptable fits were obtained for a cubic expansion of the rate constants, given by

$$\begin{aligned} \alpha_m(V) &= A_m \exp[b_{m1}(V - V_m) + c_{m1}(V - V_m)^2 + d_{m1}(V - V_m)^3]/RT \\ \beta_m(V) &= A_m \exp[b_{m2}(V - V_m) + c_{m2}(V - V_m)^2 + d_{m2}(V - V_m)^3]/RT \\ \alpha_h(V) &= A_h \exp[-[b_{h1}(V - V_h) + c_{h1}(V - V_h)^2 + d_{h1}(V - V_h)^3]/RT \\ \beta_h(V) &= A_h \exp[b_{h2}(V - V_h) + c_{h2}(V - V_h)^2 + d_{h2}(V - V_h)^3]/RT. \end{aligned} \quad (5.48)$$

The best fit of this nonlinear thermodynamic model is shown in figure 5.6 (thick solid lines) and was obtained with the following parameters: $A_m = 0.053 \text{ ms}^{-1}$, $V_m = -56 \text{ mV}$, $b_{m1} = -260$, $c_{m1} = 2.20$, $d_{m1} = 0.0052$, $b_{m2} = 64.85$, $c_{m2} = 2.02$, $d_{m2} = 0.036$, $A_h = 0.0017 \text{ ms}^{-1}$, $V_h = -80 \text{ mV}$, $b_{h1} = 163$, $c_{h1} = 4.96$, $d_{h1} = 0.062$, $b_{h2} = -438$, $c_{h2} = 8.73$, $d_{h2} = -0.057$. Figure 5.6 (thick solid lines) shows that this model could capture the form of the voltage dependence of the time constants. In particular, it

could fit the saturating values for the time constants at a depolarized level in a manner similar to the empirical functions used for the Hodgkin-Huxley-type model (figure 5.6, thin solid lines). Nonlinear expansions of higher order provided better fits, but the difference was not qualitative (not shown).

Using these rate constants with equation (5.39) produced acceptable voltage-clamp behavior, as shown in figure 5.7c. All protocols of activation (figure 5.7c1), deactivation (not shown), inactivation (not shown), and recovery from inactivation (figure 5.7c2) showed voltage-dependent behavior similar to that of the experimental data.

Markov Model of the T-Current

To illustrate the Markov representation, we have used a model of the T-current introduced by Chen and Hess (1990). This model was obtained based on voltage-clamp recordings and single-channel recordings of the T-current in fibroblasts, and the following optimal scheme was proposed (Chen and Hess, 1990):



Here, only k_a , k_d , k_f , and k_b are voltage dependent, while the other rates are constant. Thus activation occurs through one voltage-dependent step (k_a, k_d) and one voltage-independent step (k_o, k_c), the latter being rate limiting if k_a and k_d reach high values. Similarly, the inactivation occurs first through a voltage-independent step (k_i, k_{-i}), followed by a voltage-dependent transition (k_f, k_b) and a voltage-independent return to the closed state (k_r, k_{-r}).

Fitting the parameters of Markov models to experimental data is in general difficult (Cannon and D'Alessandro, 2006). It is not possible to obtain an analytical expression for both time constants and steady-state relations, owing to the too-great complexity of the model. In general, the activation and inactivation will be described by multiexponential processes with several time constants, and relating these multiple time constants to the time constants estimated from experimental data (figures 5.1–5.2) is not trivial. Rather, the parameters of Markov models are deduced from various experimental considerations. It is also possible to directly fit the Markov model to the original voltage-clamp traces by minimizing the error between the model and all experimental traces. Although in principle more accurate, this procedure is difficult to realize in practice because of the complexity of the model (eleven parameters here).

The choice of these parameters was guided by the following considerations (Chen and Hess, 1990): (1) The value of k_i must be close to the saturating value of the rate of inactivation at depolarized membrane potentials (figure 5.2c), and k_{-i} must be much smaller to ensure complete inactivation. (2) k_c must be close to the fastest activation time constants at negative potentials (figure 5.1c), while k_o must be large ($>1\text{ ms}^{-1}$) to be compatible with the short bursts of opening in single-channel recordings (Chen and Hess, 1990). (3) The sum $k_r + k_{-r}$ determines the rate of recovery from inactivation at negative membrane potentials. (4) The values of k_a and k_d are adjusted to obtain the best fit with activation and inactivation voltage-clamp recordings using a thermodynamic template with a linear dependence of the free energy on voltage:

$$k = k_0 \exp(qFV/RT), \quad (5.50)$$

where $q = 3.035$ is the net charge of a gating particle.

Since this scheme is cyclic, microscopic reversibility requires that the clockwise product of the rate constants equal the anticlockwise product, which in turn requires that the voltage dependence of k_f and k_b be the same as that of k_a and k_d . The optimal values of the rate constants were (all units are ms^{-1}):

$$\begin{aligned} k_a &= 6.4 \exp[qF(V - s)/RT] \\ k_d &= 0.000502 \exp[-qF(V - s)/RT] \\ k_f &= 16 \exp[qF(V - s)/RT] \\ k_b &= 2 \times 10^{-6} \exp[-qF(V - s)/RT] \\ k_o &= 3 \\ k_c &= 0.7 \\ k_i &= 0.036 \\ k_{-i} &= 8 \times 10^{-5} \\ k_r &= 0.001 \\ k_{-r} &= 0.003. \end{aligned} \quad (5.51)$$

Here, the parameters were adapted to recordings of the T-current in thalamic neurons. An additional parameter, $s = -5\text{ mV}$, was introduced to shift the voltage dependence to adjust the model to the thalamic T-current.

This model was simulated with the above expressions for rate constants and the T-current described by the following equation:

$$I_T = \bar{P}_{Ca}[O]G(V, Ca_o, Ca_i), \quad (5.52)$$

where $[O]$ is the fraction of channels in the open state. Simulated voltage-clamp experiments (figure 5.7d) show that the Chen and Hess model reproduced well the activation characteristics of the T-current (figure 5.7d1) as well as its slow recovery from inactivation (figure 5.7d2). However, this model did not quantitatively fit the T-current of thalamic neurons because it was based on single-channel recordings of the T-current in fibroblasts, which is different than the “neuronal” T-current (see analysis in Chen and Hess, 1990; see also Cannon and D’Alessandro, 2006). Obtaining a better Markov representation of the thalamic T-current would require constraining the model by single-channel recordings.

Comparison of the Different Models (Current Clamp)

The different models reviewed here for the T-current were compared in a current clamp. A single-compartment model of the thalamic relay neuron was generated (same parameters as in Destexhe et al., 1998b) and contained leak currents and the T-current according to the following equation:

$$c_m \frac{dV}{dt} = -g_L(V - E_L) - I_T, \quad (5.53)$$

where $c_m = 0.88 \text{ } \mu\text{F/cm}^2$ is the membrane capacitance, $g_L = 0.038 \text{ mS/cm}^2$ and $E_L = -77 \text{ mV}$ are the leak conductance and reversal potential, and I_T is the T-current as given by equation (5.12). These parameters were obtained by matching the model to thalamic neurons recorded *in vitro* (Destexhe et al., 1998b).

Using this model, the genesis of low-threshold spikes (LTSs) was monitored through return to resting potential after injection of hyperpolarizing currents. The empirical Hodgkin-Huxley-type model of the T-current generated LTSs in a grossly all-or-none fashion (figure 5.8a). The linear thermodynamic model (figure 5.8b) did not generate LTSs, which is consistent with the very small amplitude of the current evidenced earlier (figure 5.7b). On the other hand, the nonlinear thermodynamic model (figure 5.8c) and the Markov model of the T-current (figure 5.8d) presented a behavior more consistent with the Hodgkin-Huxley-type model. The peak amplitude of the LTSs was compared using different models in figure 5.8e. Although the shapes of the LTSs were not identical, Hodgkin-Huxley and nonlinear thermodynamic models produced remarkably similar peak amplitudes (filled circles and triangles in figure 5.8e). We therefore conclude that nonlinear thermodynamic models provide fits of a quality comparable to empirical Hodgkin-Huxley models, but that their form is physically more plausible.

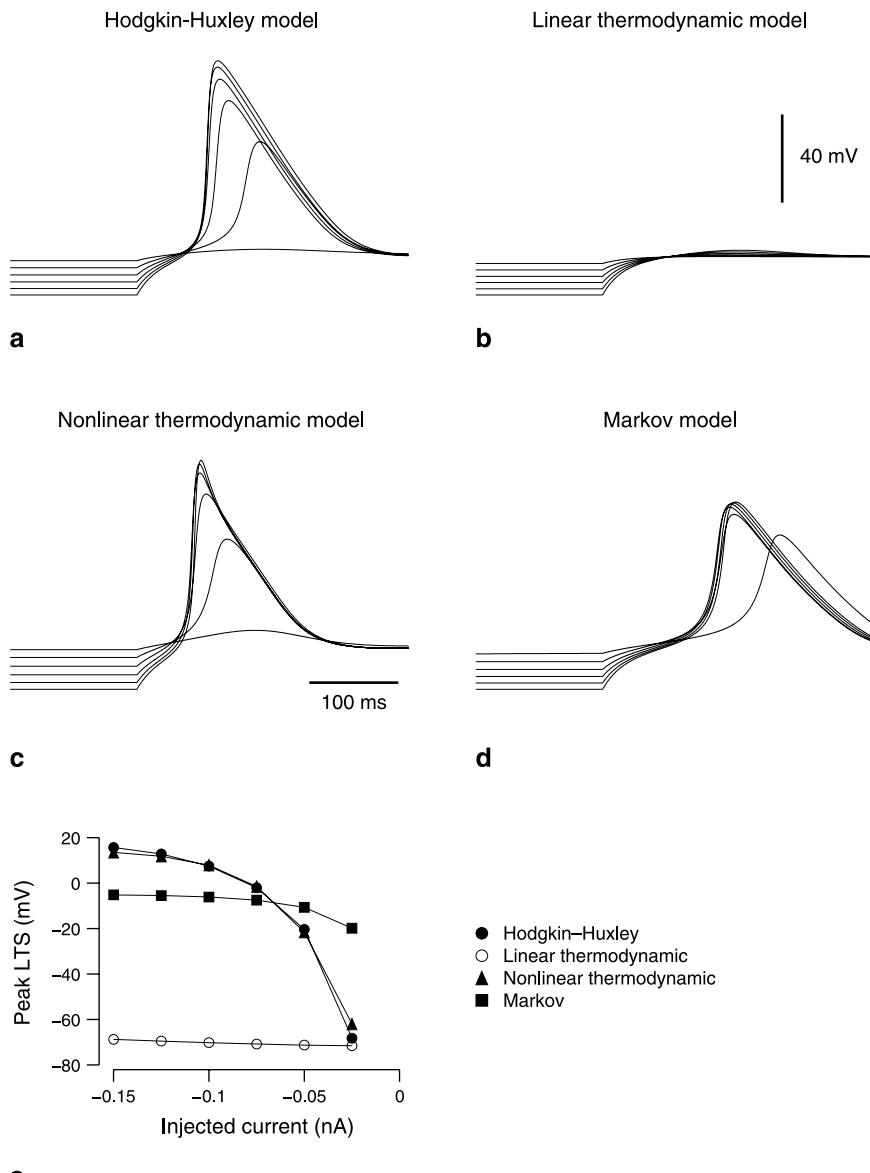


Figure 5.8

Low-threshold spikes generated by different models of the T-current. Comparison of the same current-clamp simulation for four different models of the T-current: an empirical Hodgkin-Huxley-type model (a), a linear thermodynamic model (b), a nonlinear thermodynamic model (c), and a Markov model (d). The simulation consisted in injecting hyperpolarizing current pulses of various amplitudes (-0.025 , -0.05 , -0.075 , -0.1 , -0.125 , and -0.15 nA) and of 1 s duration. At the end of the pulse, the model generated a low-threshold spike (LTS) upon return to rest. (e) Peak amplitude of low-threshold spikes generated by the different models of the T-current. All simulations were done with the same single-compartment geometry, which contained leak currents in addition to the T-current (identical parameters as in Destexhe et al., 1998). The density of T-channels was identical in all cases ($\bar{P}_{\text{Ca}} = 5 \times 10^{-5}$ cm/s) and was in the range of densities estimated from rat ventrobasal thalamic neurons (Destexhe et al., 1998a). (Figure modified from Destexhe and Huguenard, 2000.)

5.8 Conclusions

In this chapter, using concrete examples, we have illustrated several widely used formalisms to model voltage-dependent channels, and discussed how to fit the corresponding equations and parameters to experimental data (for a recent review, see Cannon and D'Alessandro, 2006).

The question of which formalism to choose for modeling voltage-dependent channels is of course entirely dependent on the type of data available and the goal of the modeling effort (see also discussion in Bruno et al., 2005). It is clear that a two-state scheme cannot capture the features of single-channel recordings, which require Markov models of sufficient complexity to account for the data. On the other hand, even simplified two- or three-state representations can capture phenomena such as action potentials (Destexhe et al., 1994). For example, if the principal requirement is to generate action potentials, it is not necessary to include all the complexity of the most sophisticated Markov diagrams of channels, and simplified representations appear sufficient. This simplistic approach may be adequate for models involving large-scale networks of thousands of cells, for which computational efficiency is a more important concern than reproducing all the microscopic features of the channels.

Finally, we would like to point to a number of resources available for modeling ion channels. The first type consists of databases freely available on the Internet. The most prominent of modeling databases is the ModelDB database (<http://senselab.med.yale.edu/senselab/ModelDB>), which provides a considerable number of program codes of published models, written using many different publicly available simulation environments. The ChannelDB database (<http://www.genesis-sim.org/hbp/channeldb/ChannelDB.html>) provides the user with a large number of ion channel models stored in simulator-independent NeuroML format. The NeuronDB database (<http://senselab.med.yale.edu/senselab/NeuronDB>) also contains useful information about the gating properties of different types of ion channels. A second type of resource is dedicated software for fitting ion channel models from experimental data. Some simulation environments, such as NEURON (see the software appendix), include automated procedures to perform such a fitting. There exist also toolboxes and free packages for fitting ion channel models to experimental data, such as NEUROFIT (<http://www.uoguelph.ca/~awillms/neurofit/>) for Hodgkin-Huxley models, or QuB (<http://www.qub.buffalo.edu/>) for Markov models. These resources are growing fast, which underlines the need for centralizing all resources in a unified database for all computational neuroscience research.

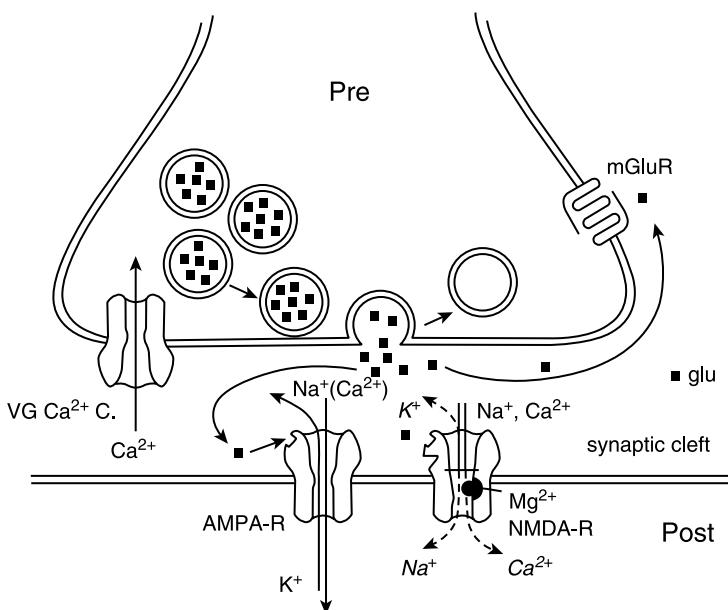
6 Modeling Synapses

Arnd Roth and Mark C. W. van Rossum

Modeling synaptic transmission between neurons is a challenging task. Since many physiological processes contribute to synaptic transmission (figure 6.1), computational models require numerous simplifications and abstractions. Furthermore, since synapses change their properties on many time scales, synaptic transmission is also a highly dynamic process. In addition, synaptic transmission is stochastic and believed to be an important source of noise in the nervous system. Finally, because the number of synapses in any decent-sized network is large (chapter 13), efficient simulation routines are important.

The human brain is estimated to contain some 10^{14} synapses. Even if all synapses were identical, modeling them would present a substantial computational challenge. Moreover, synaptic connections in the central nervous system are highly diverse. This diversity is not simply due to random variability, but most likely reflects a highly specified and precise design. For example, the same presynaptic axon can give rise to synapses with different properties, depending on the type of the postsynaptic target neuron (Thomson et al., 1993; Thomson, 1997; Reyes et al., 1998; Markram et al., 1998). At the same time, synapses on a given postsynaptic dendrite can have different properties, depending on their dendritic location and the identity of the pre-synaptic neuron (Walker et al., 2002).

Moreover, synaptic efficacies are not static but have activity-dependent dynamics in the form of short-term (Zucker and Regehr, 2002) and long-term plasticity (Abbott and Nelson, 2000), which are the consequence of complex molecular networks. Simulation of synaptic transmission therefore requires careful simplifications. This chapter describes models at different levels of realism and computational efficiency for simulating synapses and their plasticity. We also review experimental data that provide the parameter values for the synapse models, focusing on the dominant transmitter and receptor types mediating fast synaptic transmission in the mammalian central nervous system.

**Figure 6.1**

Overview of glutameric synaptic transmission. Schematic illustrating the signaling cascade underlying synaptic transmission. In response to a presynaptic action potential, calcium enters the presynaptic terminal via voltage-gated calcium channels (VG Ca²⁺ C.) and triggers the release of glutamate-containing vesicles. Glutamate diffuses into the synaptic cleft and activates postsynaptic AMPA and NMDA receptors (R), ionotropic receptors that act via opening of an ion channel permeable to sodium, potassium, and calcium, giving rise to a fast excitatory postsynaptic current (EPSC). Glutamate can also activate pre- and postsynaptic metabotropic glutamate receptors (mGluRs), which act via G-proteins. (Drawing by Henrik Alle, reproduced with permission.)

6.1 Simple Models of Synaptic Kinetics

The basic mechanism of synaptic transmission is well established: a presynaptic spike depolarizes the synaptic terminal, leading to an influx of calcium through presynaptic calcium channels, causing vesicles of neurotransmitter to be released into the synaptic cleft. The neurotransmitter binds temporarily to postsynaptic channels, opening them and allowing ionic current to flow across the membrane. Modeling this complete process is rather challenging (see sections 6.3–6.5); however, many simple phenomenological models of synapses can represent the time and voltage dependence of synaptic currents fairly well (figure 6.2).

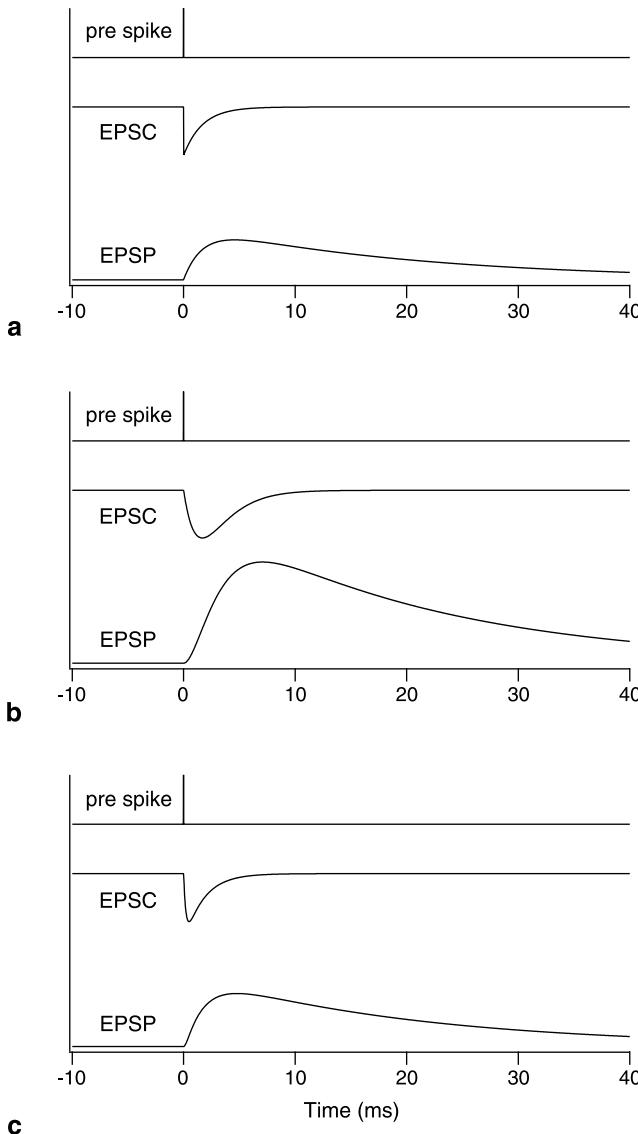


Figure 6.2

Sequence of events in simple models of synaptic kinetics. Panels a–c show the time of the presynaptic spike (top), the time course of the excitatory postsynaptic current (EPSC, middle), and the time course of the resulting postsynaptic membrane depolarization or excitatory postsynaptic potential (EPSP, bottom) for three simple models of the synaptic conductance. (a) Instantaneous rise and single-exponential decay (equation 6.1, $\tau = 1.7$ ms). (b) Alpha function (equation 6.3, $\tau = 1.7$ ms). (c) Difference of two exponentials (equation 6.4, $\tau_{\text{rise}} = 0.2$ ms, $\tau_{\text{decay}} = 1.7$ ms). The EPSC, which is negative by convention (equation 6.10), charges the membrane capacitance, leading to a transient postsynaptic depolarization whose rise time depends mainly on the kinetics of the EPSC and whose decay time constant is dominated by the postsynaptic membrane time constant. The peak synaptic conductance, postsynaptic input resistance, and membrane time constant (20 ms) were the same for all three models. EPSPs are shown for a single-compartment postsynaptic neuron. If the synapse is located on a dendrite distant from the soma, as is common, filtering by the dendritic cable will reduce the amplitude and prolong the time course of the somatic EPSP (chapter 10, section 3).

Instantaneous Rise and Single-Exponential Decay

One of the simplest and often-used models assumes an instantaneous rise of the synaptic conductance $g_{\text{syn}}(t)$ from 0 to \bar{g}_{syn} (the peak conductance) at time t_0 followed by an exponential decay with a time constant τ (figure 6.2a):

$$g_{\text{syn}}(t) = \bar{g}_{\text{syn}} e^{-(t-t_0)/\tau}. \quad (6.1)$$

For times $t < t_0$ before the onset of the synaptic conductance, $g_{\text{syn}}(t) = 0$. The time t_0 is the time of the presynaptic spike, or more realistically, a transmission delay can be included (see section 6.2).

A possible justification for this simple model is as follows: Suppose one has a large number of two-state synaptic channels. Now assume that the release of neurotransmitter, its diffusion across the cleft, the receptor binding, and channel opening all happen very quickly, so that the channels instantaneously jump from the closed to the open state. The transmitter transient disappears quickly and the open probability, and hence the conductance, decay exponentially as the channels make transitions back into the closed state. The corresponding differential equation is

$$\tau \frac{dg_{\text{syn}}(t)}{dt} = -g_{\text{syn}}(t) + \bar{g}_{\text{syn}} \delta(t_0 - t), \quad (6.2)$$

where $\delta(t)$ is the Dirac delta function. The decay time constant τ is largely determined by the unbinding of the neurotransmitter from the receptor channel. In particular for certain types of inhibitory postsynaptic currents (IPSCs), the exponential synapse is a good approximation because the rising phase is much shorter than their decay phase (e.g., Kraushaar and Jonas, 2000), but also for fast AMPA-mediated excitatory postsynaptic currents (EPSCs), it is a decent model.

Alpha Function

However, for most synapses, the rising phase of synaptic conductances has a finite duration, which can have strong effects on network dynamics (van Vreeswijk et al., 1994). The so-called alpha function describes a conductance that has a rising phase that is not infinitely fast, but has a certain rise time (figure 6.2b):

$$g_{\text{syn}}(t) = \bar{g}_{\text{syn}} \frac{t - t_0}{\tau} e^{1-(t-t_0)/\tau}. \quad (6.3)$$

However, because it has just a single time constant, τ , the time courses of the rise and decay are correlated and cannot be set independently. In general this condition is not physiologically realistic; nevertheless the alpha function provides a reasonable approximation for some synapses. It peaks at a time $t - t_0 = \tau$, and for the above definition the peak amplitude is given directly by \bar{g}_{syn} . As before, $g_{\text{syn}}(t) = 0$ for $t < t_0$.

Difference of Two Exponentials

A more general function describing synaptic conductance profiles consists of a difference of two exponentials, one for the rising and one for the decay phase. It allows these time constants to be set independently ($\tau_{\text{rise}} \neq \tau_{\text{decay}}$), so that for $t \geq t_0$ (figure 6.2c):

$$g_{\text{syn}}(t) = \bar{g}_{\text{syn}} f(e^{-(t-t_0)/\tau_{\text{decay}}} - e^{-(t-t_0)/\tau_{\text{rise}}}). \quad (6.4)$$

The normalization factor f is included to ensure that the amplitude equals \bar{g}_{syn} . The conductance peaks at a time:

$$t_{\text{peak}} = t_0 + \frac{\tau_{\text{decay}}\tau_{\text{rise}}}{\tau_{\text{decay}} - \tau_{\text{rise}}} \ln\left(\frac{\tau_{\text{decay}}}{\tau_{\text{rise}}}\right), \quad (6.5)$$

and the normalization factor for the amplitude follows as

$$f = \frac{1}{-e^{-(t_{\text{peak}}-t_0)/\tau_{\text{rise}}} + e^{-(t_{\text{peak}}-t_0)/\tau_{\text{decay}}}}. \quad (6.6)$$

In the case of most synapses, rapid binding is followed by slow unbinding of the transmitter. This particular profile of the synaptic conductance can be interpreted as the solution of two coupled linear differential equations (Wilson and Bower, 1989; Destexhe et al. 1994):

$$g_{\text{syn}}(t) = \bar{g}_{\text{syn}} f g(t) \quad (6.7)$$

$$\frac{dg}{dt} = -\frac{g}{\tau_{\text{decay}}} + h \quad (6.8)$$

$$\frac{dh}{dt} = -\frac{h}{\tau_{\text{rise}}} + h_0\delta(t_0 - t), \quad (6.9)$$

with h_0 a scaling factor. The alpha function is retrieved in the limit when both time constants are equal.

The time course of most synaptic conductances can be well described by this difference of two exponentials. It is of course possible to further increase the number of exponentials describing the synaptic conductance to obtain even better fits, but then no closed-form expression for the time of peak and the normalization factor for the amplitude exists. Furthermore, extracting a large number of exponentials from noisy data is cumbersome (see later discussion).

Conductance-Based and Current-Based Synapses

Most ligand-gated ion channels mediating synaptic transmission, such as AMPA-type glutamate receptors and γ -aminobutyric acid type A (GABA_A) receptors (see

section 6.3), display an approximately linear current-voltage relationship when they open. They can therefore be modeled as an ohmic conductance g_{syn} which when multiplied with the driving force, the difference between the membrane potential V and the reversal potential E_{syn} of the synaptic conductance, gives the synaptic current

$$I_{\text{syn}} = g_{\text{syn}}(t)[V(t) - E_{\text{syn}}]. \quad (6.10)$$

However, in some cases, in particular in analytical models, it may be a useful approximation to consider synapses as sources of current and not a conductance, i.e., without the dependence on the membrane potential V as described by equation (6.10). This can be achieved, for example, by using a fixed value $V = V_{\text{rest}}$ in equation (6.10). For small excitatory synapses on a large compartment, this is a good approximation. In that case, the depolarization of the membrane will be small and hence the difference between V and E_{syn} will hardly change during the excitatory postsynaptic potential (EPSP). However, if the synapse is located on a thin dendrite, the local membrane potential V changes considerably when the synapse is activated (Nevian et al., 2007). In that case a conductance-based synapse model seems more appropriate. Nevertheless, voltage-dependent ion channels in the dendritic membrane, together with a voltage-dependent component in the synaptic conductance g , mediated by NMDA receptors (see next), can at least partially compensate for the change in membrane potential and can cause synapses to effectively operate as current sources (Cook and Johnston, 1999).

For inhibitory synapses, the distinction between conductance-based and current-based models is particularly important because the inhibitory reversal potential can be close or even above the resting potential. As a result, the resulting synaptic current becomes highly dependent on the postsynaptic voltage. This shunting leads to shortening of the membrane time constant, which does not occur for current-based models. This can have substantial consequences for network simulations (Koch, 1999; Vogels and Abbott, 2005; Kumar et al., 2008; chapter 13, section 7).

Simple Descriptions of NMDA Receptor-Mediated Synaptic Conductances

Excitatory synaptic currents commonly have both AMPA and NMDA components. The NMDA current is mediated by NMDA channels, which like AMPA channels, are activated by glutamate but have a different sensitivity. The NMDA receptor-mediated conductance depends on the postsynaptic voltage, i.e., equation (6.10) is not valid. The voltage dependence is due to the blocking of the pore of the NMDA receptor from the outside by a positively charged magnesium ion. The channel is nearly completely blocked at resting potential, but the magnesium block is relieved if the cell is depolarized. The fraction of channels $u(V)$ that are not blocked by magnesium can be fitted to

$$u(V) = \frac{1}{1 + e^{-aV}[\text{Mg}^{2+}]_o/b}, \quad (6.11)$$

with $a = 0.062 \text{ mV}^{-1}$ and $b = 3.57 \text{ mM}$ (Jahr and Stevens, 1990). Here $[\text{Mg}^{2+}]_o$ is the extracellular magnesium concentration, usually 1 mM. If we make the approximation that the magnesium block changes instantaneously with voltage and is independent of the gating of the channel, the net NMDA receptor-mediated synaptic current is given by

$$I_{\text{NMDA}} = g_{\text{NMDA}}(t)u(V)[V(t) - E_{\text{NMDA}}], \quad (6.12)$$

where $g_{\text{NMDA}}(t)$ could for instance be modeled by a difference of exponentials, equation (6.4). Thus NMDA receptor activation requires both presynaptic activity (to provide glutamate) and postsynaptic activity (to release the magnesium block). This property has wide implications for the functional role of NMDA receptors for synaptic integration (Schiller et al., 2000; Losonczy and Magee, 2006; Nevian et al., 2007), neuronal excitability (Rhodes and Llinás, 2001), network dynamics (Compte, 2006; Durstewitz and Seamans, 2006; Durstewitz and Gabriel, 2007) and synaptic plasticity (Bliss and Collingridge, 1993).

Finally, NMDA receptors have a significant permeability for calcium ions and constitute one of the pathways for calcium entry, which is relevant for synaptic plasticity (Nevian and Sakmann, 2006). To calculate the calcium current through the NMDA receptor, the Goldman-Hodgkin-Katz equation should be used (see equation 4.2 and Badoual et al., 2006).

Synaptic Time Constants for AMPA, NMDA, and GABA_A

The time constants of synaptic conductances vary widely among synapse types. However, some general trends and typical values can be identified. First, synaptic kinetics tends to accelerate during development (T. Takahashi, 2005). Second, synaptic kinetics becomes substantially faster with increasing temperature. In the following discussion, we therefore focus on data from experiments performed at near-physiological temperature (34–38 °C). This is important because the temperature dependence is different for the various processes involved in synaptic transmission (transmitter diffusion, receptor kinetics, single channel conductance, etc.). In analogy with voltage-gated channels, the temperature dependence can be approximated with a Q_{10} factor, which describes the speedup with every 10 degrees of temperature increase (equation 5.43). In particular, the temperature dependence of synaptic kinetics is generally steep, with a Q_{10} of typically around 2–3 (Huntsman and Huguenard, 2000; Postlethwaite et al., 2007). Further complication arises because the Q_{10} of the different transitions in the state diagram can vary (e.g., Cais et al., 2008), so that a uniform scaling of all time constants is not appropriate.

AMPA receptor-mediated EPSCs at glutamatergic synapses are among the fastest synaptic currents, but because different types of neurons express different subtypes of AMPA receptors (see section 6.3), a range of time constants is observed. The fastest AMPA receptor-mediated EPSCs are found in the auditory system where $\tau_{\text{decay}} = 0.18$ ms in the chick nucleus magnocellularis (Trussell, 1999). Excitatory synapses onto interneurons in the cortex and hippocampus also tend to have a fast AMPA component ($\tau_{\text{rise}} = 0.25$ ms and $\tau_{\text{decay}} = 0.77$ ms in dentate gyrus basket cells; Geiger et al., 1997). Synapses onto pyramidal neurons tend to have slower AMPA components ($\tau_{\text{rise}} = 0.2$ ms and $\tau_{\text{decay}} = 1.7$ ms in neocortical layer 5 pyramidal neurons; Häusser and Roth, 1997b).

The NMDA receptor-mediated component of the EPSC is typically more than an order of magnitude slower than the AMPA receptor-mediated component. The slow unbinding rate of glutamate from the NMDA receptor makes NMDA more sensitive to glutamate and causes glutamate to stick longer to the receptor, increasing the open time. The decay time constants at near-physiological temperature range from 19 ms in dentate gyrus basket cells (Geiger et al., 1997) to 26 ms in neocortical layer 2/3 pyramidal neurons (Feldmeyer et al., 2002), and up to 89 ms in CA1 pyramidal cells (Diamond, 2001). Likewise NMDA rise times, which are about 2 ms (Feldmeyer et al., 2002), are slower than AMPA rise times. The reversal potential of both AMPA and NMDA receptor-mediated currents is, conveniently, around 0 mV under physiological conditions.

Synapses often contain both AMPA and NMDA receptor-mediated conductances in a specific ratio (Feldmeyer et al., 2002; Losonczy and Magee, 2006). However, there are also NMDA receptor-only synapses, so-called silent synapses (Liao et al., 1995; Isaac et al., 1995; Groc et al., 2006). These have little or no synaptic current at the resting potential because the NMDA receptors are blocked by magnesium. A few types of synapses are AMPA receptor-only synapses during development (Cathala et al., 2000; Clark and Cull-Candy, 2002).

GABA_A receptor-mediated IPSCs tend to decay more slowly than the AMPA conductances. GABAergic synapses from dentate gyrus basket cells onto other basket cells are faster: $\tau_{\text{rise}} = 0.3$ ms and $\tau_{\text{decay}} = 2.5$ ms (Bartos et al., 2001) than synapses from basket cells to granule cells: $\tau_{\text{rise}} = 0.26$ ms and $\tau_{\text{decay}} = 6.5$ ms (Kraushaar and Jonas, 2000). Reversal potentials of GABA_A receptor-mediated conductances change with development (Ben-Ari, 2002) and activity (Fiumelli and Woodin, 2007).

Experimentally, accurate measurement of synaptic currents and time constants is difficult because the voltage in dendrites is difficult to control (known as the space-clamp problem; chapter 10, section 3), leading to an overestimate of the rise and decay time constants of the synaptic conductance at distal dendritic locations. Care should therefore be taken to minimize this by optimizing recording conditions, recording from small (i.e., electrically compact) cells, or employing analysis tech-

niques that provide an unbiased estimate of the decay time constant of the synaptic conductance (Häusser and Roth, 1997b; Williams and Mitchell, 2008).

6.2 Implementation Issues

Given the large number of synapses in most network simulations, efficient simulation of synapses is key. For home-brewed simulations, an easy trick is the following: Whereas updating all the synaptic conductances at every time step according to, for instance equation (6.1), is computationally costly, if the postsynaptic neuron contains just a single compartment (e.g., an integrate-and-fire neuron, chapter 7), all synaptic conductances with identical parameters can be lumped together into a total synaptic conductance (Lytton, 1996). This total synaptic conductance g_{Σ} increases when any of the synapses onto the neuron are activated, i.e., $g_{\Sigma} = g_{\Sigma} + g_i$, where i sums over the activated synapses. Only the decay of this summed conductance has to be calculated at each time step dt . For instance, in the case of a single exponential decay, $g_{\Sigma}(t + dt) = g_{\Sigma}(t) \exp(-dt/\tau)$, meaning that the computational cost of updating the synaptic conductance is on the order of the number of neurons; i.e., it is comparable to the cost of updating the neuron's membrane potentials only.

Implementation in Simulator Packages

Models of synaptic transmission have also been implemented in neural simulation packages (see the software appendix). Work is under way to improve the accuracy and efficiency of these implementations and, depending on the application, different simulation strategies may be the most accurate and most efficient (Brette et al., 2007). One strategy is to compute the synaptic conductance time courses as the solutions of the corresponding linear differential equations, which can be integrated numerically using discrete time steps, or in some cases can be solved analytically (Destexhe et al., 1998a; Rotter and Diesmann, 1999; Carnevale and Hines, 2006). Instead of analytical solutions, precomputed lookup tables for the synaptic and neuronal dynamics can be also used (Ros et al., 2006).

Another approach is to use event-driven methods, which are particularly efficient in cases that allow an analytical solution for the response of the postsynaptic neuron (Brette, 2006; Carnevale and Hines, 2006; Rudolph and Destexhe, 2006; chapter 7). These algorithms extrapolate the time at which neuron will spike, removing the need for simulating the intermediate time steps. However, if input rates are high, these extrapolations have to be revised frequently, nullifying their benefit. Finally, hybrid strategies that combine time-driven and event-driven methods are available (Morrison et al., 2007). Most synapse models discussed here have also been implemented in hardware, both using analog very large-scale integration (VLSI) (Rasche and

Douglas, 1999; Mira and Álvarez, 2003; Zou et al., 2006; Bartolozzi and Indiveri, 2007) and field-programmable gate arrays (FPGAs) (Guerrero-Rivera et al., 2006).

Axonal, Synaptic, and Dendritic Delays

The postsynaptic response to a presynaptic spike is not instantaneous. Instead, axonal, synaptic, and dendritic delays all contribute to the onset latency of the synaptic current with respect to the time of the presynaptic spike. Since these delays can add up to several milliseconds and thus can be comparable in duration to the synaptic time constant, they can have consequences for synchronization and oscillations (Brunel and Wang, 2003; Maex and De Schutter, 2003; chapter 13, section 3) and the stability of spike timing-dependent plasticity (STDP) in recurrent networks (Morrison et al., 2007). In practice, long delays benefit parallel simulation algorithms (chapter 13, section 6).

6.3 Biophysics of Synaptic Transmission and Synaptic Receptors

After describing these simplified models of synapses, we now focus on the underlying mechanisms of synaptic transmission, which will allow discussion of the more realistic models presented in section 6.4. The precise shape of synaptic events is determined by the biophysical mechanisms that underlie synaptic transmission. Numerous in-depth reviews of synaptic physiology and kinetic models of synaptic receptor molecules exist (Jonas and Spruston, 1994; Jonas, 2000; Attwell and Gibb, 2005; Lisman et al., 2007).

Transmitter Release

The signaling cascade underlying synaptic transmission (figure 6.1) begins with the arrival of the presynaptic action potential in the presynaptic terminal, where it activates various types of voltage-gated ion channels (Meir et al., 1999), in particular, voltage-gated calcium channels. The calcium current through these channels reaches its maximum during the repolarization phase of the action potential as the driving force for calcium ions increases while the calcium channels begin to deactivate (Borst and Sakmann, 1998). This calcium influx leads to a local increase in the intracellular calcium concentration (see chapter 4, section 3 and section 6.5), which in turn triggers the release of neurotransmitter molecules from vesicles, owing either to their full fusion with the presynaptic membrane or the formation of a fusion pore (He et al., 2006).

Transmitter Diffusion

Following its release, the neurotransmitter diffuses in the synaptic cleft. The spatio-temporal profile of transmitter concentration in the synaptic cleft depends on a num-

ber of factors, the first being the rate of release of presynaptic vesicles (figure 6.1). In the absence of full fusion, the release rate of transmitter molecules is also limited by the diameter of the fusion pore (Stiles et al., 1996). The subsequent movement of transmitter molecules is determined by their diffusion coefficient in the synaptic cleft (Nielsen et al., 2004), the geometry of the synapse, and the reuptake of transmitter by transporter molecules in presynaptic cells and glia. Synaptic geometry can change during development (Cathala et al., 2005) and can depend on activity (Genoud et al., 2006), which in turn can change the time course of transmitter in the synaptic cleft. The density of transporter molecules tends to increase during development, speeding up the reuptake of transmitter and sharpening the transmitter transient (Diamond, 2005). The number of transporter molecules available to bind transmitter also depends on past synaptic activity.

Receptor Binding

During the brief transient increase of transmitter concentration in the synaptic cleft, some of the transmitter molecules bind to receptor molecules in the postsynaptic membrane, causing conformational changes that lead to the transient opening of ion channels coupled to the receptors (Madden, 2005; Mayer and Armstrong, 2004; Mayer, 2005, 2006). The kinetics of the postsynaptic receptors is an important determinant of the time course of the synaptic conductance. Functional AMPA, NMDA, and GABA_A receptors are composed of several subunit proteins. The functional properties of a receptor strongly depend on its subunit composition (Geiger et al., 1995; Cull-Candy and Leszkiewicz, 2004; Paoletti and Neyton, 2007). The heterogeneity of receptor subunit composition can explain a large part of the variability in the time course of the synaptic conductance within and between different classes of synaptic connections. Furthermore, during development, a switch in subunit expression occurs in many types of synapses, leading to changes in postsynaptic receptor properties. In particular, such subunit switches tend to accelerate the kinetics of the receptors during development (Feldmeyer and Cull-Candy, 1996; Joshi et al., 2004; T. Takahashi, 2005). Receptor properties can also be changed by various accessory and modulatory subunits. Prominent examples are transmembrane AMPAR regulatory proteins (TARPs) such as stargazin, which modulates AMPA receptor gating and trafficking by distinct domains (Tomita et al., 2005). Thus for a realistic model, the functional properties of synaptic receptors should be studied under conditions that are as close as possible to their native state and environment at the synapse (DiGregorio et al., 2007).

The kinetics of synaptic receptors and their associated ion channels can be described by Markov models (chapter 5, section 5). At any given time, the receptor-channel complex is assumed to exist in a given state, such as closed, open, or desensitized. Rate constants determine the transitions between any two of these states.

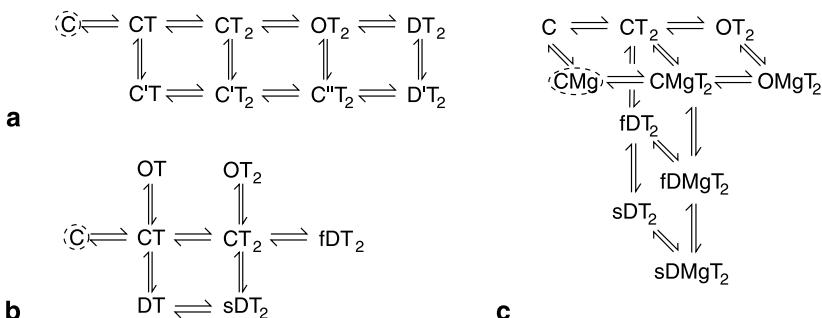


Figure 6.3

Kinetic schemes for common synaptic channels. Closed states are denoted with a C, open states with an O, and desensitized states with a D. Primes are used to distinguish otherwise identical states. The transmitter is denoted by T. A circled state denotes the state at rest, in the absence of a transmitter. (a) AMPA receptor kinetic model after DiGregorio et al. (2007). Two transmitter molecules T need to bind for the channel to open. (b) GABA_A receptor kinetic model after Pugh and Raman (2005). Note that there are two open states in this case. The desensitized states are split into slow (sD) and fast (fD) ones. (c) NMDA receptor kinetic model after Kampa et al. (2004). The model includes the kinetics of magnesium (Mg) binding. For simplicity, the binding with both glutamate molecules has been captured in a single transition. Again there are both slow and fast desensitized states.

State transitions involving the binding of a transmitter molecule to the receptor are described by rate constants that are proportional to the transmitter concentration. The rate constants are free parameters in these models, which need to be extracted from experimental data. Except for the simplest Markov models with only two or three states, which might not describe the synaptic receptors very well, the number of rate constants and thus the number of free parameters is significantly larger than the number of free parameters in simple functions describing synaptic conductance kinetics (section 6.1). It is usually necessary to expose receptors to a number of protocols involving both short and long pulses of transmitter concentration (e.g., Häusser and Roth, 1997a) to adequately constrain a Markov model of their kinetics. A number of Markov models for AMPA receptors from different types of neurons have been obtained in this way (Nielsen et al., 2004). Figure 6.3a shows a kinetic model of synaptic AMPA receptors in cerebellar granule cells (DiGregorio et al., 2007) and table 6.1 provides the rate constants for the models. Kinetic models of NMDA receptors with a slow component of the magnesium unblock (Kampa et al., 2004; Vargas-Caballero and Robinson, 2003, 2004; figure 6.3c) can be useful for simulations of NMDA receptor-dependent synaptic plasticity (section 6.7).

In order to use Markov models of synaptic receptors, a model of the time course of the transmitter concentration in the synaptic cleft is needed to drive the Markov model. Models of transmitter diffusion in detailed synaptic geometries (section 6.5) may provide fairly accurate predictions of the time course of transmitter concentration, but are computationally very expensive. However, because the transmitter

Table 6.1

Rate constants for the models in figure 6.3

Transition	Forward	Backward
AMPA		
C-CT	13.66 mM ⁻¹ ms ⁻¹	8.16 ms ⁻¹
CT-CT ₂	6.02 mM ⁻¹ ms ⁻¹	4.72 ms ⁻¹
CT-C'T	0.11 ms ⁻¹	0.1 ms ⁻¹
C'T-C'T ₂	13.7 mM ⁻¹ ms ⁻¹	1.05 ms ⁻¹
C'T ₂ -C''T ₂	0.48 ms ⁻¹	0.98 ms ⁻¹
CT ₂ -OT ₂	17.2 ms ⁻¹	3.73 ms ⁻¹
C''T ₂ -D'T ₂	10.34 ms ⁻¹	4 ms ⁻¹
CT ₂ -C'T ₂	2 ms ⁻¹	0.19 ms ⁻¹
OT ₂ -C''T ₂	0.0031 ms ⁻¹	0.0028 ms ⁻¹
OT ₂ -DT ₂	0.11 ms ⁻¹	0.09 ms ⁻¹
DT ₂ -D'T ₂	0.0030 ms ⁻¹	0.0013 ms ⁻¹
GABA _A		
C-CT	0.04 μM ⁻¹ ms ⁻¹	2 ms ⁻¹
CT-OT	0.03 ms ⁻¹	0.06 ms ⁻¹
CT-DT	0.000333 ms ⁻¹	0.0007 ms ⁻¹
CT-CT ₂	0.02 μM ⁻¹ ms ⁻¹	4 ms ⁻¹
CT ₂ -OT	10 ms ⁻¹	0.4 ms ⁻¹
CT ₂ -sDT ₂	1.2 ms ⁻¹	0.006 ms ⁻¹
CT ₂ -fDT ₂	15 ms ⁻¹	0.15 ms ⁻¹
DT-sDT ₂	0.015 μM ⁻¹ ms ⁻¹	0.007 ms ⁻¹
NMDA ^a		
C-CT ₂	10 μM ⁻¹ s ⁻¹	5.6 s ⁻¹
CMg-CMgT ₂	10 μM ⁻¹ s ⁻¹	17.1 s ⁻¹
CT ₂ -OT ₂	10 s ⁻¹	273 s ⁻¹
CMgT ₂ -OMgT ₂	10 s ⁻¹	548 s ⁻¹
CT ₂ -fDT ₂	2.2 s ⁻¹	1.6 s ⁻¹
CMgT ₂ -fDMgT ₂	2.1 s ⁻¹	0.87 s ⁻¹
fDT ₂ -sDT ₂	0.43 s ⁻¹	0.50 s ⁻¹
fDMgT ₂ -sDMgT ₂	0.26 s ⁻¹	0.42 s ⁻¹
O-OMg (at +40 mV)	0.05 μM ⁻¹ s ⁻¹	12,800 s ⁻¹
x-xMg (at +40 mV)	50.10 ⁻⁶ μM ⁻¹ s ⁻¹	12.8 s ⁻¹

^aSee also <http://senselab.med.yale.edu/modedb>, model = 50207.

transient is much faster than the receptor dynamics, a brief square pulse or a single-exponential decay suffices (Destexhe et al., 1998a).

Second-Messenger Synapses

In the preceding discussion, the transmitter binds to the ion channels directly, so-called ionotropic receptors. However, this is not always the case. In second-messenger or metabotropic receptors (Coutinho and Knöpfel, 2002), the neurotransmitter binds to a receptor that activates an intermediate G-protein, which in turn binds to a G-protein-gated ion channel and opens it. As a result of the intermediate cascade, the synaptic responses of second-messenger synapses are usually slower, but they can also be very sensitive and highly nonlinear. For the same reason, unified models of second-messenger synapses are rare and can display a rich behavior. The best-known examples of these synapses are the metabotropic glutamate receptors and the inhibitory receptor GABA_B . In the case of GABA_B , the inhibitory current becomes disproportionately stronger with multiple events (Destexhe et al., 1998a). The nonlinearity of the mGluR receptor in retinal bipolar cells has, for instance, been proposed to filter out photoreceptor noise (van Rossum and Smith, 1998). Finally, mGluR receptors have also been found presynaptically (T. Takahashi et al., 1996; R. J. Miller, 1998).

6.4 Modeling Dynamic Synapses

Many of the biophysical mechanisms involved in synaptic transmission are use dependent (Zucker and Regehr, 2002). For example, residual elevations of the presynaptic calcium concentration that are due to a previous action potential can increase the probability of subsequent transmitter release. An increase in the synaptic response to successive stimuli is called synaptic facilitation. On the other hand, depletion of the pool of readily releasable vesicles can cause a decrease of the synaptic response to successive stimuli, called short-term synaptic depression (to distinguish it from long-term synaptic depression, LTD, the counterpart of long-term potentiation, LTP). Depression and facilitation of synaptic currents can also have postsynaptic components (Heine et al., 2008). Cumulative desensitization of postsynaptic receptors can contribute to synaptic depression, while increased occupancy of transporters can lead to a prolonged presence of transmitter molecules in the synaptic cleft and a facilitation of synaptic responses.

Short-term synaptic plasticity is a major determinant of network dynamics. It can provide gain control, underlie adaptation and detection of transients, and constitute a form of short-term memory (Abbott and Regehr, 2004; chapter 13, section 3). The characteristics of short-term synaptic plasticity depend on (and can be used to define) the type of synaptic connection in a neural circuit (Silberberg et al., 2005). A number

of functional consequences of short synaptic dynamics have been suggested: temporal filtering (Puccini et al., 2007), gain adaptation (Abbott et al., 1997), decorrelation of inputs (Goldman et al., 2002), working memory (Mongillo et al., 2008), and adaptive visual processing (van Rossum et al., 2008).

The amplitude and time constants of synaptic depression and facilitation, which can be determined experimentally using trains of synaptic stimuli, are free parameters of simple phenomenological models of short-term synaptic plasticity (Abbott et al., 1997; Markram et al., 1998; Tsodyks et al., 2000). These models are a good approximation of the short-term dynamics of many types of synapses and can be implemented efficiently.

We present here the model of Tsodyks and Markram as described in Tsodyks et al. (1998, 2000). In this model, the “resources” of the synapse that provide the synaptic current or conductance can exist in one of three states: the recovered or resting state, with occupancy x ; the active or conducting state, y ; and an inactive state, z . At any given time

$$x + y + z = 1. \quad (6.13)$$

A fourth state variable, u , describes the “use” of synaptic resources in response to a presynaptic action potential arriving at time $t = t_0$. Its dynamics is given by

$$\frac{du}{dt} = -\frac{u}{\tau_{\text{facil}}} + U(1-u)\delta(t-t_0), \quad (6.14)$$

so u is increased by $U(1-u)$ with each action potential and decays back to zero with a time constant τ_{facil} . This u drives the dynamics of the other three state variables according to

$$\frac{dx}{dt} = \frac{z}{\tau_{\text{rec}}} - ux\delta(t-t_0) \quad (6.15)$$

$$\frac{dy}{dt} = -\frac{y}{\tau_{\text{decay}}} + ux\delta(t-t_0) \quad (6.16)$$

$$\frac{dz}{dt} = \frac{y}{\tau_{\text{decay}}} - \frac{z}{\tau_{\text{rec}}}, \quad (6.17)$$

where τ_{rec} is the time constant of recovery from depression and τ_{decay} (corresponding to τ in equation 6.1) is the time constant of decay of the synaptic current or conductance, which is proportional to y .

Typical observed time constants for recovery are between 100 and 1,000 ms *in vitro*. The time constants and magnitude of synaptic depression *in vivo* have not been clearly established, however. With background activity, synapses might already

be in a somewhat depressed state before stimulation begins, making depression less pronounced than in a slice preparation, for which background activity is commonly absent (Reig et al., 2006). Furthermore, synaptic depression is also known to depend on the specific synaptic connection (Thomson and Lamy, 2007) and brain area (Wang et al., 2006). Models with multiple time constants have also been proposed (Varela et al., 1997) and biophysical models have been made (Hennig et al., 2008).

An efficient scheme to simulate synaptic depression is discussed in Morrison et al. (2008). For an example implementation in NEURON, see model 3815 in ModelDB (<http://senselab.med.yale.edu/ModelDb>); other efficient implementations of related models can be found in Giugliano et al. (1999) and Giugliano (2000). A hardware implementation in analog VLSI is described by Bartolozzi and Indiveri (2007).

6.5 Stochastic Models of Synaptic Transmission

The descriptions of synaptic transmission we have discussed so far are deterministic. However, synaptic responses show considerable trial-to-trial variability, owing to the stochastic nature of many steps in the signaling cascade underlying synaptic transmission (for a review, see Faisal et al., 2008).

Sources of variability have been studied in nearly all components of the cascade and include variations in the waveform of the presynaptic action potential and stochastic gating of presynaptic calcium channels, which lead to variability in the amount of calcium entering the presynaptic terminal in response to a presynaptic action potential; randomness of the diffusion of the individual calcium ions; stochastic gating of the release mechanism for presynaptic vesicles; variability across vesicles in the number of transmitter molecules contained in or released from a vesicle, owing to variations in vesicle size, the degree of filling with transmitter, and incomplete emptying in the absence of full fusion; variability in the location of the release site in the presynaptic membrane with respect to the postsynaptic density; random diffusion of transmitter molecules in the synaptic cleft; stochastic gating of the postsynaptic receptors; and changes in the number of available postsynaptic receptors that are due to lateral diffusion of receptors in the postsynaptic membrane (Heine et al., 2008).

In principle, a composition of biophysically realistic quantitative descriptions of the stochasticity of all these steps can be put together, but this would result in a complicated model with many parameters that are difficult to estimate from available data. In practice it may be more useful to represent several steps in a simpler, phenomenological stochastic model. Here we focus on simple stochastic descriptions of transmitter release (for a biophysical model of presynaptic calcium dynamics, see Meinrenken et al., 2002), stochastic modeling of transmitter diffusion (see also chapter 3), and stochastic models of channel gating.

Stochastic Models of Vesicle Release

The description of synaptic potentials and synaptic currents as a sum of probabilistically generated “quantal components” has its origin in the work of del Castillo and Katz (1954) at the neuromuscular junction. More recently, it has been used to interpret the fluctuations of EPSP amplitudes measured in paired recordings from synaptically connected neurons in the central nervous system (Thomson et al., 1993; Markram et al., 1997; Brémaud et al., 2007). In its simplest version, the synapse is described as an arrangement of n independent release sites, each of which releases a quantum of transmitter with a probability p in response to a presynaptic action potential. Each released quantum of transmitter generates a contribution of size q to the postsynaptic response r (which can be a synaptic conductance, current, or postsynaptic potential amplitude). Therefore r follows a binomial distribution and the mean response $\langle r \rangle$ is

$$\langle r \rangle = npq. \quad (6.18)$$

Individual responses can be simulated using binomially distributed random numbers. If n is low, this can be done by generating n random deviates uniformly distributed between zero and one. The number of deviates smaller than p gives the synaptic amplitude after multiplication by q . By using different values of p_i and q_i , this scheme can be generalized to the case that different release sites (indexed by i) have different release probabilities p_i and quantal sizes q_i .

Typical values for p are between 0.1 and 0.5 but can vary widely, whereas n typically ranges between 3 and 20 (Markram et al. 1997; Brémaud et al., 2007). The number of anatomical contacts is probably close to n , but because combining anatomy and physiology in a single experiment is difficult, a precise quantification is currently lacking.

For a model that combines stochastic release and short-term plasticity, see Maass and Zador (1999). It is generally believed that the stochastic release of vesicles is a large source of variability in the nervous system. Whether this variability is merely the consequence of the underlying biophysics or has some functional role is a topic of active debate.

Stochastic Models of Transmitter Diffusion

A simulator for the stochastic simulation of transmitter diffusion and receptor binding is MCell (see the software appendix). MCell simulates diffusion of single molecules and the reactions that take place as transmitter molecules bind to receptors and transporters, which can be used to quantify the different sources of variability (Franks et al., 2003; Raghavachari and Lisman, 2004).

Stochastic Models of Receptors

A stochastic description of the stochastic opening of the postsynaptic receptors is straightforward once a Markov model has been formulated. Instead of using the average transition rates in the state diagram, at each time step the transitions from one state to another are drawn from a binomial distribution, with a probability p equal to the transition rate per time step, and n equal to the number of channels in the state at which the transitions are made. The fewer receptors there are in the postsynaptic membrane, the more pronounced the fluctuations around the average behavior will be. The typical number of postsynaptic receptors is believed to be in the range of 10 to 100. Compared with the other sources of variability, the noise from stochastic opening of the receptor channels is usually minor (van Rossum et al., 2003). Nevertheless, the fluctuations across trials can be used to determine the unitary conductance of the receptors using so called nonstationary noise analysis (Hille, 2001). Computationally, these simulations are expensive because the number of transitions is assumed to be small in the simulation, whereas the transition rates can be high, necessitating a very small time step.

6.6 Modeling Synaptic Learning Rules

One of the central hypotheses in neuroscience is that learning and memory are reflected in the weight of synapses. Many experiments and models have studied therefore how synapses change in response to neural activity, so-called activity-dependent synaptic plasticity. Some models have as their aim the construction of learning rules so that the network performs a certain task, such as learning input-output associations (for instance, using backpropagation), independent component analysis, or achieving sparse codes. Because the links between these models and the physiology are often tentative, we will not describe them here. Another class of models, discussed here, uses a phenomenological approach and implements biologically observed learning rules to examine their functional consequences.

The biological pathways underlying synaptic plasticity are only just starting to be understood, but it clear that they contain many ingredients and regulation mechanisms. Although efforts are being made to model at this level (chapter 3), for network studies one often prefers a simpler model. This separation of levels of complexity is often desirable conceptually, but also from a computational perspective. In plasticity studies one is often interested in a network's properties as they evolve over longer periods of time or when many stimuli are presented. This means that simulation studies of learning are computationally particularly expensive and require efficient algorithms.

The fundamental experimental finding is that brief exposure to high pre- and postsynaptic activity causes long-lasting increases in synaptic strength, which can persist up to months *in vivo* (Bliss and Lomo, 1973). This finding is consistent with Hebb's hypothesis (Hebb, 1949) and is called Hebbian learning. This phenomenon is known as LTP. Similarly low, but nonzero, levels of activity can cause long-lasting weakening of synapses, LTD. Consistent with most experiments, increases and decreases in synaptic efficacy in most models are implemented through changes in the postsynaptic conductance \bar{g}_{syn} , although in some studies evidence for changes in the release probability was found and has been included in models (Tsodyks and Markram, 1997; Maass and Zador, 1999).

Rate-Based Plasticity

In an effort to condense the experimental data on plasticity, early plasticity models used often a rate-based formulation, which, for instance, takes the form (Sejnowski, 1977):

$$\Delta w_{ij} = \varepsilon(r_i - a)(r_j - b), \quad (6.19)$$

where r_i is the presynaptic firing rate, r_j is the postsynaptic firing rate, w_{ij} is the synaptic weight connecting the neurons, and a and b are constants. Finally, the proportionality constant ε is often taken to be a small number, ensuring that the synaptic weights change only slowly. The rates are assumed to be averaged over times longer than any single neuron dynamics (such as spikes) but shorter than the stimulus dynamics. Such a learning rule, for instance, can be used to train networks receiving sensory input in an unsupervised manner.

It is not difficult to see that this learning rule diverges, however, because once a synapse becomes stronger, it will evoke a higher postsynaptic activity, which leads to a further increase in its strength, etc. A variety of solutions have been proposed to deal with this problem. These include imposing maximum and minimum weights, normalizing the sum of all the weights coming into a neuron, while the Bienenstock-Cooper-Munro (BCM) rule adjusts the thresholds of LTP and LTD as a function of average postsynaptic activity (Bienenstock et al. 1982; Dayan and Abbott, 2002). The precise solution strongly affects the resulting behavior of the learning rule while at the same time experimental data on this issue are largely lacking. Despite this uncertainty, or maybe owing to the freedom it allowed, many rate-based plasticity models have been used successfully to describe memory models (Bienenstock et al., 1982; K. D. Miller, 1996; Miikkulainen and Sirosh, 2005).

Spike Timing-Dependent Plasticity

Changes in synaptic weight depend not only on mean firing rates but also on the precise temporal order of pre- and postsynaptic spikes, a phenomenon known as STDP

(Abbott and Gerstner, 2004; Caporale and Dan, 2008). Given a spike at time t_i in the presynaptic neuron and at time t_j in the postsynaptic neuron, a simple description of STDP is

$$\Delta w_{ij} = \epsilon \operatorname{sign}(t_i - t_j) \exp(-|t_i - t_j|/\tau_{\text{STDP}}), \quad (6.20)$$

where τ_{STDP} is the time window for spike pair interaction, some 20 ms. Note that the weight change is discontinuous when $t_i = t_j$.

However, the apparent simplicity of the STDP rule is a bit misleading because for an actual implementation, additional specifications are needed. First, without any further constraints, weights will diverge as more and more spike pairs are encountered. As for rate-based plasticity rules, bounds can be imposed. These can be hard bounds that simply cap the maximal weight, or softer bounds that, for instance, reduce the amount of potentiation as the weight increases. In addition, homeostatic mechanisms can be included that modify the learning rules based on the total postsynaptic activity. Second, one needs to define which spike pairs contribute to weight changes. For instance, what happens if a presynaptic spike is followed by two postsynaptic spikes; do both contribute to a weight modification? Such rules can be classified and analyzed formally to some extent (Burkitt et al., 2004).

Although in equation (6.20) the STDP rule is independent of the pre- and postsynaptic rate, experiments show that higher pairing frequencies usually cause LTP and lower frequencies cause LTD. As long as there is no experimental evidence that STDP and rate-based learning are separate plasticity mechanisms, it is convenient to unify them in a single framework. Unified models of STDP and rate-based learning that even include BCM-like plasticity have been created by allowing higher-order interactions between spikes (Pfister and Gerstner, 2006).

This raises the question of whether the temporal character of STDP matters at all or is simply a side effect of a biological implementation of rate-based learning. Although this issue has not been settled yet, there is evidence that some perceptual learning shares properties with STDP (Dan and Poo, 2006) and that some plasticity phenomena can only be explained by using STDP (Young et al., 2007).

Analysis of STDP in networks is difficult because of the mutual interaction between the spike patterns in the network, which will modify the synapses, and the plasticity, which will modify the activity patterns. Even creating a stable asynchronous state in a network with STDP is challenging but possible (Morrison et al., 2007).

Biophysical Models of Long-Term Potentiation

The molecular biology of LTP is quite complicated and involves numerous pathways. LTP is critically dependent on calcium influx in the postsynaptic membrane that activates CaMKII (Lisman et al., 2002). The calcium influx is strongly depen-

dent on spike sequence and has been used as a basis for STDP (Shouval et al., 2002; Badoual et al., 2006). Experiments have further distinguished between different phases of plasticity, so-called early LTP (lasting a few hours) and late LTP (perhaps lasting indefinitely). In particular, the stability of LTP has received attention from modelers (Graupner and Brunel, 2007; Lisman and Raghavachari, 2006).

Again, synapses of different neurons obey different learning rules and even on a given neuron, synapses at different dendritic locations can have distinct plasticity rules (Sjöström et al., 2008). Thus far this level of complexity has largely been ignored in functional models.

6.9 Conclusions

We have presented synaptic models at various levels of physiological detail and realism. We started with some quite simple formulations that nevertheless provide decent models for the synaptic current. Increases in computer speed and increased understanding of neural transmission have made more and more complex models possible. The researcher will need to consider a tradeoff between complex models and simpler models that may ignore subtleties. However, a simple heuristic model is not necessarily inferior, because it might actually do a better job in describing the physiology than a complicated model with incorrect parameters.

Unless biophysical mechanisms of synaptic transmission themselves are the subject of study, mechanistic detail is usually not needed. Experimental data that constrain all parameters of these phenomenological models, such as voltage-clamp recordings of synaptic current waveforms, are more likely to be available than full experimental characterizations of the functional properties of the molecules and the biophysical mechanisms involved. Furthermore, complex models not only require more computer time but also more time to set up, verify, tune, and analyze. An informed choice of model will depend on the question that is being asked. No matter what model is used, parameter values should be considered carefully. For instance, the strong temperature dependence of gating kinetics implies that data recorded at room temperature should not simply be plugged into an *in vivo* model.

Recent years have also seen increased understanding of issues such as long-term and short-term synaptic plasticity, receptor diffusion, and subunit composition, to name but a few. Unfortunately for the modeler, the known diversity among synapses and their properties has also increased. The number of variations in synaptic subunit composition, modulation (not discussed here), and long- and short-term dynamics is ever increasing, and there is no reason to expect a slowdown in this trend. The rules that underlie this diversity are so far largely unknown, so intuition and exploration are required from the modeler.

7

Modeling Point Neurons: From Hodgkin-Huxley to Integrate-and-Fire

Nicolas Brunel

This chapter describes why and in which conditions simplified spiking neurons are useful. It then introduces the prototypical simplified spiking neuron model, the leaky integrate-and-fire (LIF) neuron. Next, several models that are intermediate between Hodgkin-Huxley (HH)-type neurons and LIF neurons are presented. The idea of such models is to capture some dynamic features of real neurons that are not present in LIF neurons while keeping the model as simple as possible. The resulting models can have additional nonlinearities and/or an additional variable compared with the LIF neuron. The chapter ends by describing several ways by which one can obtain parameters of simplified models from intra- or extracellular recordings.

7.1 Why Integrate-and-Fire-Type Neurons?

Computational neuroscience literature abounds with point-spiking neuron models. At one extreme, there are models that use the Hodgkin-Huxley formalism (described in chapter 5, section 2). These models are typically composed of a variable number of ionic currents: two in the case of the original Hodgkin-Huxley model (Hodgkin and Huxley, 1952) and up to ten or more as, e.g., in models of cell types as a thalamic relay cell (McCormick and Huguenard, 1992), a motoneuron (Purvis and Butera, 2005), or a hippocampal pyramidal cell (Borg-Graham, 1999). Ionic currents are described mathematically in terms of activation and/or inactivation variables. The advantage of these models is that they can be used to quantitatively fit the dynamics of real neurons, following the pioneering work of Hodgkin and Huxley. The drawback is that they are highly nonlinear and defined by a large number of variables, which makes mathematical analysis of their behavior extremely difficult and makes numerical simulations of large networks of such neurons very expensive computationally.

At the other extreme, there is the integrate-and-fire (IF) neuron, introduced one century ago (Lapicque, 1907; Stein, 1965; Brunel and van Rossum, 2007). The IF

neuron is the simplest possible spiking neuron described by a membrane potential variable with a stable resting membrane potential. Because of this simplicity, it is unable to reproduce the wide diversity of behaviors seen in real neurons. On the other hand, this very simplicity is what has allowed theorists to study it analytically in great detail, especially in the presence of stochastic inputs (Tuckwell, 1988). Furthermore, it also enables theorists to study network dynamics analytically, and its simplicity allows numerical simulations of large networks of such neurons. Simulations of networks of tens or even hundreds of thousands of neurons (described in chapter 13) are today routinely feasible in a reasonable time (see also Brette et al., 2007). The LIF model is described in detail in section 7.3.

A very broad class of models falls in between these two extremes. The idea of such models is to simplify the Hodgkin-Huxley formalism while keeping dynamic behaviors of interest that are not present in the IF model. The goal is (1) to better understand such dynamic behaviors by studying the most mathematically compact model that exhibits them and (2) to obtain the computationally cheapest (and possibly mathematically analyzable) model to use in network studies. Several models that belong to that class are discussed in section 7.4 and listed here:

- IF-like spiking models with nontrivial subthreshold dynamics (e.g., with subthreshold resonance behavior): the resonate-and-fire neuron (Izhikevich, 2001) and generalized integrate-and-fire (GIF) neuron (Richardson et al., 2003)
- two-variable excitable models: FitzHugh-Nagumo (Fitzhugh, 1961; Nagumo et al., 1962) and Morris-Lecar (Morris and Lecar, 1981)
- one-variable excitable models, such as the quadratic (QIF) or theta model (Ermentrout and Kopell, 1986; Ermentrout, 1996), and the exponential IF (EIF) model (Fourcaud-Trocmé et al., 2003)
- IF models with firing rate adaptation (ALIF) (Treves, 1993, Ermentrout, 1998; Meunier and Borejsza, 2005)
- integrate-and-fire-or-burst (IFB) model (Smith et al., 2000)

This review of models will end with the Izhikevich model, a two-variable model that can reproduce many of the dynamics mentioned here, depending on the values taken by the parameters defining the model (Izhikevich, 2003, 2004, 2006). Izhikevich (2004) also gives an account of the tradeoff between biological realism and computational efficiency for various models.

For the sake of brevity, only point neuron models that are described by at most two coupled first-order differential equations are treated here. Hence, several simplified spiking neuron models that do not fall in this category will not be described in this chapter. Some popular examples of models that are not discussed include Gerstner's spike response model (SRM), which is defined in terms of two kernels describ-

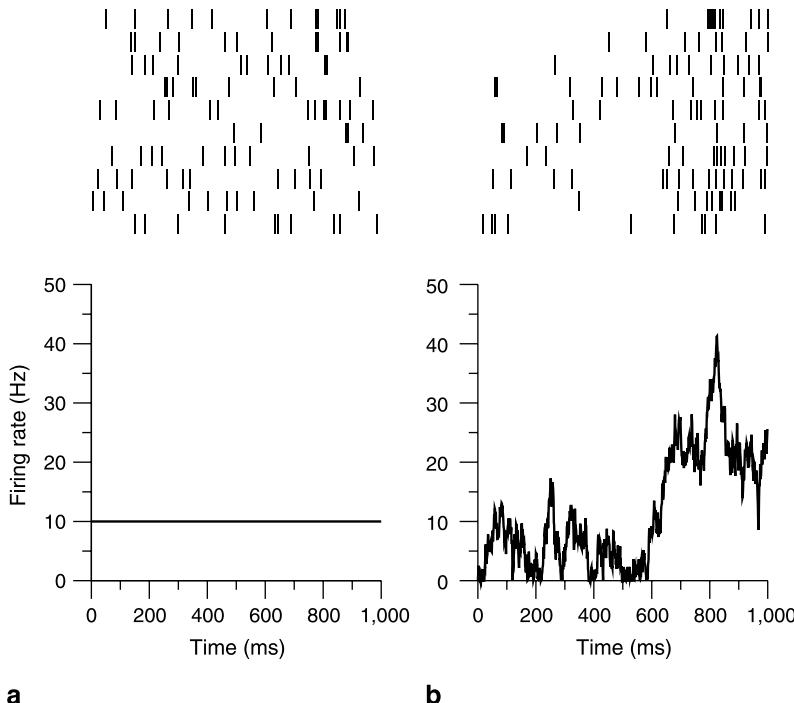
ing the refractoriness properties and the effect of inputs—hence it is in fact a broad class of models encompassing several of the ones discussed earlier (Gerstner, 1995; Gerstner and Kistler, 2002); the Hindmarsh-Rose model of a bursting neuron, which is a three-variable model (Hindmarsh and Rose, 1984); another three-variable model introduced by Rotstein et al. (2006) to account for the dynamics of stellate cells of the entorhinal cortex; and models described by maps such as the one introduced by Rulkov (2002). Two-compartment integrate-and-fire neurons are also not discussed (J. Feng and Li, 2002; Larkum et al., 2004; Capaday and van Vreeswijk, 2006).

7.2 Stochastic Point Process Models

Before introducing the integrate-and-fire neuron, it is useful to consider another class of simplified spiking neuron models that do not have a variable representing the membrane potential. Rather, these models take the spike train as being a realization of a stochastic point process—typically an inhomogeneous Poisson or gamma process—whose stochastic intensity (instantaneous firing rate) is a function of current synaptic inputs. Several good reviews about such point process models are available in the literature (e.g., D. H. Johnson, 1996), so these models are described only briefly here.

Point processes are characterized by a stochastic intensity $v(t, t_i)$ that describes the instantaneous probability of emitting a spike. The stochastic intensity depends in general on time t and the previous history of the process, i.e., on previous spike times t_i . An important class of point process models that are often used in neuronal modeling is the class of renewal processes, in which the firing probability depends only on the current time and the time of the last spike t_{last} . A special member of that class is the inhomogeneous Poisson process, in which the firing probability is independent of the times of previous spikes. If the process does not depend explicitly on time, it is said to be stationary.

In particular, a stationary Poisson process has a time-independent stochastic intensity v . The interspike interval (ISI) distribution for such a process is the exponential distribution $P(T) = v \exp(-vT)$. Generating a realization of a Poisson process is simple: one generates a sequence of interspike intervals by generating a sequence of uncorrelated random numbers according to the ISI distribution (see chapter 2, section 7 for random number generation). This procedure can be easily generalized to any renewal process. One simply replaces the exponential distribution by the ISI distribution of the corresponding process. A simple generalization is a Poisson process with a refractory period τ_{rp} , i.e., $P(T) = 0$ for $T < \tau_{rp}$, $P(T) = v \exp[-vT/(1 - v\tau_{rp})]/(1 - v\tau_{rp})$. Another popular example is the gamma process, defined by an ISI distribution of the form $P(T) = (kv)^k T^{k-1} \exp(-kvT)/\Gamma(k)$ where $k > 0$

**Figure 7.1**

Realizations of Poisson processes with stationary (a) or nonstationary (b) stochastic intensity (instantaneous firing rate). In both cases the top panel shows ten realizations of the process, while the bottom panel shows the stochastic intensity versus time.

and Γ is the gamma function (Abramowitz and Stegun, 1970). This distribution, for integer k , corresponds to the distribution of the sum of k successive ISIs of a Poisson process. Hence, for $k = 1$, we recover the Poisson process, while for a large k the distribution becomes more and more peaked around the mean interspike interval.

Generating a realization of a nonstationary point process (i.e., with a time-varying stochastic intensity v) is trickier. A simple but inefficient approach is to discretize time in bins dt that are much smaller than the time scale at which the intensity is varying. The time bin dt should also satisfy $v(t) dt \ll 1$ at all times; then in each time bin dt a spike is present with a probability $v(t) dt$. Much more efficient methods are described in Devroye (1986) and D. H. Johnson (1996). Several realizations of a homogeneous and an inhomogeneous Poisson process are shown in figure 7.1.

Although these models have been useful for describing the dynamics of real neurons, they are inadequate for understanding the mechanisms giving rise to specific firing patterns. Thus we now turn to integrate-and-fire neurons and their variants.

7.3 The Leaky Integrate-and-Fire Neuron

The leaky integrate-and-fire neuron (Lapicque, 1907; Knight, 1972; Tuckwell, 1988) is the simplest model described by a membrane potential variable with the following features: a membrane time constant, a stable resting potential, a membrane potential threshold for spike generation, a reset for membrane potential after a spike is emitted, and a refractory period.

The Model

The model is characterized by its membrane potential V , which behaves as a passive integrator of synaptic inputs in the subthreshold range:

$$c \frac{dV}{dt} = -g_L(V - E_L) + I_{\text{syn}}(t), \quad (7.1)$$

where c is the neuron capacitance, g_L is the leak conductance, E_L is the leak (or resting) potential, and $I_{\text{syn}}(t)$ are the synaptic inputs. The equation for membrane potential dynamics is often rewritten as

$$\tau_m \frac{dV}{dt} = -(V - E_L) + \tilde{I}_{\text{syn}}(t), \quad (7.2)$$

where $\tau_m = c/g_L$ is the membrane time constant and $\tilde{I}_{\text{syn}} = I_{\text{syn}}/g_L$ describes the synaptic inputs in units of voltage. In the following discussion, for the sake of convenience we will drop the \sim sign, and I will denote inputs in units of voltage or currents, depending on the context.

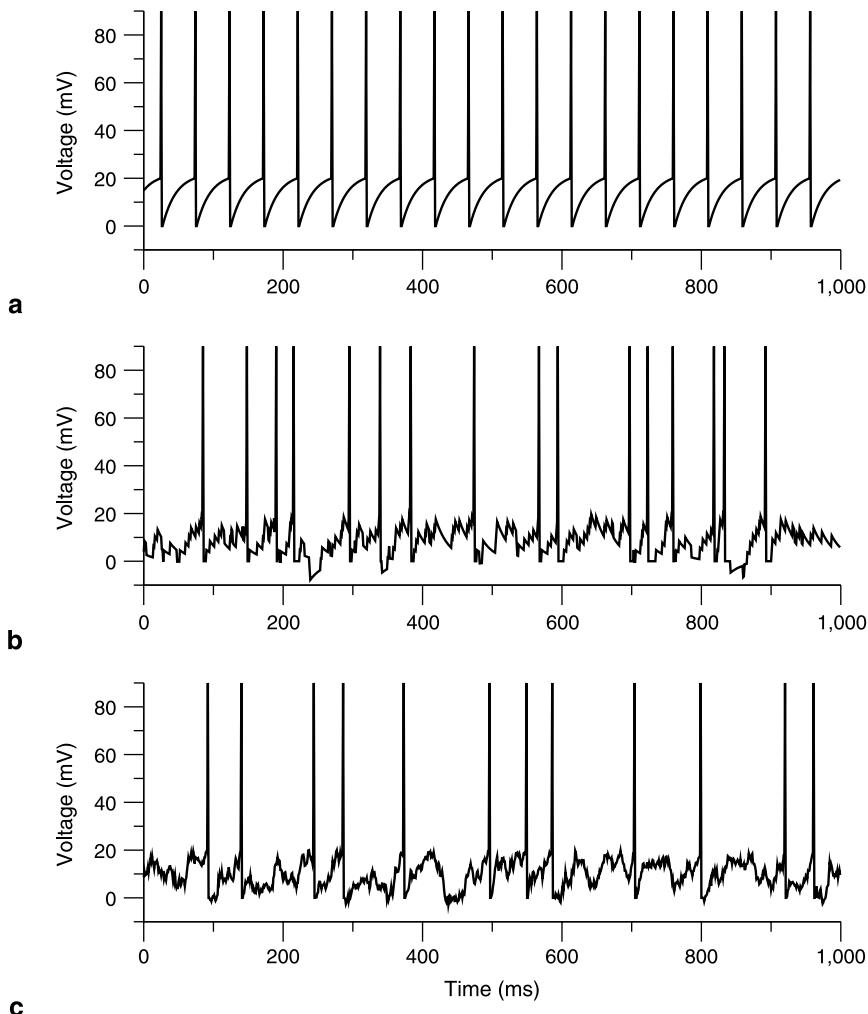
When the voltage reaches a threshold V_t , the neuron emits a spike and the voltage is reset to a reset potential V_r after an absolute refractory period τ_{rp} . The dynamics of the model is shown in figure 7.2 in response to various input currents.

Note that an even simpler model with no leak current has been proposed (the so-called perfect integrate-and-fire neuron) (Knight, 1972). Another variant of the LIF model has a voltage-independent negative leak, with an absorbing lower bound on voltage so that it cannot diverge to arbitrary hyperpolarized values (Fusi and Mattia, 1999).

The LIF model is particularly useful because one can compute its firing rate and even higher-order moments of its interspike interval distribution in several contexts, including stochastic ones. We now discuss these response properties in some detail.

Response to Constant Currents

The simplest case is when one has a constant input current I ; it is then easy to show that the firing rate v is given by

**Figure 7.2**

Dynamics of an LIF neuron in response to various types of input currents. Parameters: $E_L = 0$, $V_t = 20$ mV, $V_r = 0$ mV, $\tau_m = 20$ ms, $\tau_{sp} = 2$ ms. (a) Constant input. (b) Inputs are a combination of excitatory and inhibitory Poisson spike trains. (c) White noise input.

$$v = \frac{1}{\tau_{rp} + \tau_m \log\left(\frac{I-V_r}{I-V_t}\right)}. \quad (7.3)$$

It exhibits a sharp threshold for spike generation at $I = V_t$; above that threshold the firing rate grows as $1/\{\tau_m \log[(I - V_t)/(V_t - V_r)]\}$. We will see in the next section that this behavior is nongeneric and disappears when a more realistic spike-generation behavior is introduced. For larger values of the current, the f - I curve (see section 7.4 and figure 7.7) has an approximately linear behavior:

$$v = \frac{I}{\tau_m(V_t - V_r)}. \quad (7.4)$$

Finally, the firing rate saturates for large currents at a frequency that is equal to the inverse of its refractory period, $v \sim 1/\tau_{rp}$.

Stochastic Input Currents

When the synaptic inputs are stochastic, the behavior of the model becomes much harder to analyze. However, one can perform the so-called diffusion approximation, which consists in approximating the synaptic inputs by a constant term I plus a white noise term with a variance density σ^2 , provided the following conditions are satisfied:

- Individual synaptic currents have small amplitudes compared with the threshold.
- Synapses are activated by independent Poisson processes.
- Synaptic currents are much shorter than the membrane time constant so one can effectively neglect their time constant.

After this simplification is done, it is possible to compute

- the stationary firing rate
- the coefficient of variation (CV) of interspike intervals
- the Laplace transform of the ISI distribution
- the distribution of membrane potential
- the linear response (susceptibility) of the instantaneous firing rate

For more details, the interested reader may consult Ricciardi (1977), Tuckwell (1988), Brunel and Hakim (1999), Brunel (2000), Brunel et al. (2001), Lindner and Schimansky-Geier (2001), and the recent reviews by Burkitt (2006a, 2006b).

These studies provide a rich picture of the dynamics of LIF neurons in stochastic conditions. In the presence of fluctuating inputs, the LIF neurons can be in two qualitatively distinct regimes. When the average inputs are below threshold, the LIF neuron has a fluctuation-driven firing at low rates and thus its spike train is typically

very irregular (CV close to 1). In this regime, the LIF model has an ISI density close to that of a Poisson process. On the other hand, when the average inputs are above threshold, the LIF neuron fires in a close to periodic fashion, with variability in ISI intervals around the mean induced by those fluctuations. The distribution of membrane potentials is also quite different in both cases. With subthreshold inputs, the distribution is similar to a Gaussian, except close to the threshold, where the distribution vanishes. On the other hand, in the limit of large suprathreshold inputs, the distribution tends to be uniform between reset and threshold. Finally, the linear firing rate response is also quite distinct in both regimes; whereas the response to oscillatory inputs decays as a function of frequency in the subthreshold regime, it exhibits resonances at the mean firing rate (and possibly higher harmonics if firing is very regular) in the suprathreshold regime.

7.4 From HH-Type Neurons to IF-Type Neurons

The Hodgkin-Huxley formalism has been highly successful in terms of quantitatively reproducing the dynamics of real neurons (see chapter 5 for more details). Here we explore several ways in which HH-type models can be reduced to simplified models that, however, keep some nontrivial features of the dynamics of the original model. We start by describing how to incorporate subthreshold phenomena that are not present in the LIF model (the GIF model). We then present models that describe generation of an action potential in a simplified way compared with the Hodgkin-Huxley model: two-variable models such as the FitzHugh Nagumo, which describe spike generation as well as repolarization following the spike, and one-variable nonlinear models (NLIF) that describe spike generation but still need a sharp LIF-type reset to account for postspike repolarization. The next step is to present simplified models that account for spike-frequency adaptation (ALIF) and bursting (IFB). Finally, we describe nonlinear two-variable models that can account for most of the phenomena mentioned here, depending on the value of the parameters defining the model.

Subthreshold Dynamics: The GIF Model

Many neurons have subthreshold dynamics that are richer than that of the LIF neuron. Subthreshold dynamics is sometimes characterized in real neurons by the impedance. This impedance characterizes how voltage is affected by an oscillatory input current with a frequency f (see, e.g., Koch, 1999; Hutcheon and Yarom, 2000; Richardson et al., 2003). Two types of impedance curves (shown schematically in the insets of figure 7.3) have been reported in real neurons: either a monotonically decaying function of frequency (similar to the LIF model), or a peak around a

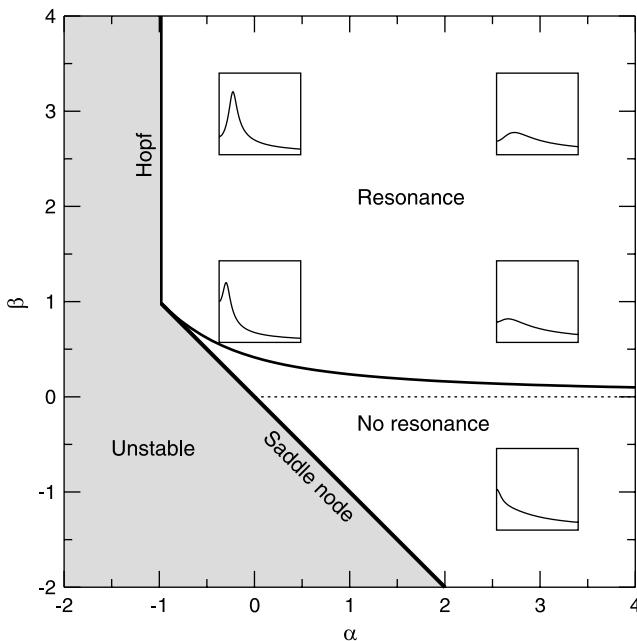


Figure 7.3

Qualitative behaviors of the impedance of the GIF model in the α - β plane. The gray region is the region in which the model has no stable fixed point. The fixed point can destabilize either through a saddle-node bifurcation (a real eigenvalue becoming positive) or a Hopf bifurcation (two complex conjugate eigenvalues acquiring a positive real part). In the stable region, the model exhibits either subthreshold resonance (above the solid line) or not (below the solid line). On the dotted line, at $\beta = 0$, $\alpha > 0$ the model is identical to a leaky integrate-and-fire neuron. The insets show the impedance versus frequency curve at representative points on the plane. Note that the peak in the impedance curve becomes more pronounced as one comes closer to the Hopf bifurcation line. (Modified from Richardson et al., 2003.)

positive frequency (subthreshold resonance). The latter behavior is present in the Hodgkin-Huxley model in its original version (Hodgkin and Huxley, 1952), but also in numerous other models (Hutcheon and Yarom, 2000).

To understand how this subthreshold resonance behavior comes about, it is useful to linearize Hodgkin-Huxley equations around a subthreshold holding potential (Koch, 1999; Hutcheon and Yarom, 2000; Richardson et al., 2003). This means expanding the voltage as $V = V_0 + v$. The sodium activation is expanded as $m = m_0 + m^{\infty\prime}(V_0)w_m$, where m_0 is the equilibrium value of sodium activation at $V = V_0$, $m^{\infty\prime}(V_0)$ is the slope of the steady-state activation curve at $V = V_0$, and w_m describes the deviation of the sodium activation around its equilibrium value at V_0 . Likewise, we expand the sodium inactivation as $h = h_0 + h^{\infty\prime}(V_0)w_h$, and the potassium activation as $n = n_0 + n^{\infty\prime}(V_0)w_n$. Other voltage-gated currents are treated in the same fashion.

The linearized equations are then

$$c \frac{dv}{dt} = -gv - \sum_{a=m,h,n,\dots}^n \tilde{g}_a w_a + I_{\text{syn}}(t) \quad (7.5)$$

$$\tilde{\tau}_a \frac{dw_a}{dt} = -w_a + v, \quad (7.6)$$

where

$g = g_L + \sum_a g_a m_{a,0}$ is the effective input conductance at $V = V_0$;

$\tilde{g}_a = g_a(V_0 - E_a)m_a^{\infty}(V_0)$ is the effective conductance associated with the corresponding activation or inactivation variable at $V = V_0$; and

$\tilde{\tau}_a = \tau_a(V_0)$ is the time constant of the activation or inactivation variable at $V = V_0$.

Equations (7.5) and (7.6) are much simpler than the original set of equations; however they still form a system of $n + 1$ equations, where n is the number of activation and inactivation variables of the model we started from. One is often able to further simplify this system of equations by taking into account the fact that some variables have time scales that are much faster than others (e.g., the activation variable of sodium currents) and/or their associated effective conductance is negligible (this is the case for variables that are far from their half-activation point).

In many models (as, e.g., models with a hyperpolarization-activated cation current (I_h) when the voltage is sufficiently hyperpolarized), such a procedure leaves us with a single activation or inactivation variable. The dynamics of the resulting model can be written as

$$c \frac{dv}{dt} = -gv - g_1 w + I_{\text{syn}}(t) \quad (7.7)$$

$$\tau_1 \frac{dw}{dt} = -w + v, \quad (7.8)$$

where g_1 measures the strength of the feedback provided by the variable w on the voltage v . In the case $g_1 > 0$, the variable acts as a negative feedback on voltage. Inactivation of sodium and calcium currents, activation of potassium currents, and activation of the H-current all belong to this category. On the other hand, in the case of $g_1 < 0$, the variable acts as a positive feedback on voltage. This is what happens with activation of sodium and calcium currents, and with inactivation of potassium currents.

Rescaling time with τ_1 , and dividing equation (7.7) by c/τ_1 makes it apparent that the model is described by two dimensionless parameters only: an effective leak $\alpha = g\tau_1/c$ and an effective coupling $\beta = g_1\tau_1/c$ (Richardson et al., 2003):

$$\frac{dv}{dt'} = -\alpha v - \beta w + I'_{\text{syn}}(t) \quad (7.9)$$

$$\frac{dw}{dt'} = -w + v, \quad (7.10)$$

where both time and currents have been rendered adimensional, $t' = t/\tau_1$, $I'_{\text{syn}}(t) = I_{\text{syn}}(t)\tau_1/c$. Note that in the case of $\beta = 0$ we recover the usual integrate-and-fire model. This model has a stable equilibrium at $v = 0$, $w = 0$ when $\alpha = -1$ and $\alpha + \beta > 0$. On the line $\alpha = -1$, $\beta > 1$, this fixed point becomes unstable through a Hopf bifurcation, while on the line $\alpha + \beta > 0$, $\alpha > -1$ it becomes unstable through a saddle-node bifurcation (see chapter 1, section 3). In the following discussion we focus on the stable region. We now consider the response of such a model to currents that are often used to probe subthreshold dynamics: oscillatory currents and current steps.

Response to Oscillatory Currents

In response to an oscillatory current $I_{\text{syn}}(t) = I_1 \cos(\omega t')$, the system responds as $V = V_1(\omega) \cos[\omega t' + \phi(\omega)]$, where

$$\frac{V_1(\omega)}{I_1} = \sqrt{\frac{(1 + \omega^2)}{(\alpha + \beta - \omega^2)^2 + \omega^2(1 + \alpha)^2}} \quad (7.11)$$

$$\phi(\omega) = \text{atan} \left[\omega \frac{\beta - (1 + \omega^2)}{\beta + \alpha(1 + \omega^2)} \right]. \quad (7.12)$$

Depending on the values of α and β , the impedance $V_1(\omega)/I_1 h$ has two qualitatively distinct shapes: for $\beta < [(\alpha + 1)^2 + 1]^{1/2} - (1 + \alpha)$ (lower white region in figure 7.3), the impedance decays monotonically with frequency (as in the LIF model); while in the opposite region $\beta > [(\alpha + 1)^2 + 1]^{1/2} - (1 + \alpha)$ (upper white region in figure 7.3) the impedance has a peak at a frequency

$$f_R = \frac{1}{2\pi\tau_1} \sqrt{\sqrt{(\alpha + \beta + 1)^2 - (\alpha + 1)^2} - 1}. \quad (7.13)$$

Response to Current Steps

In response to a current step, the membrane potential response also has qualitatively different time courses, depending on α and β (sketched in the insets of figure 7.4): for $\beta > (\alpha - 1)^2/4$, it exhibits damped oscillations before reaching its equilibrium value; for $\alpha > 1$ and $0 < \beta < (\alpha - 1)^2/4$, it goes through a single maximum (or minimum for a hyperpolarizing current) before reaching its equilibrium value (overshoot, or

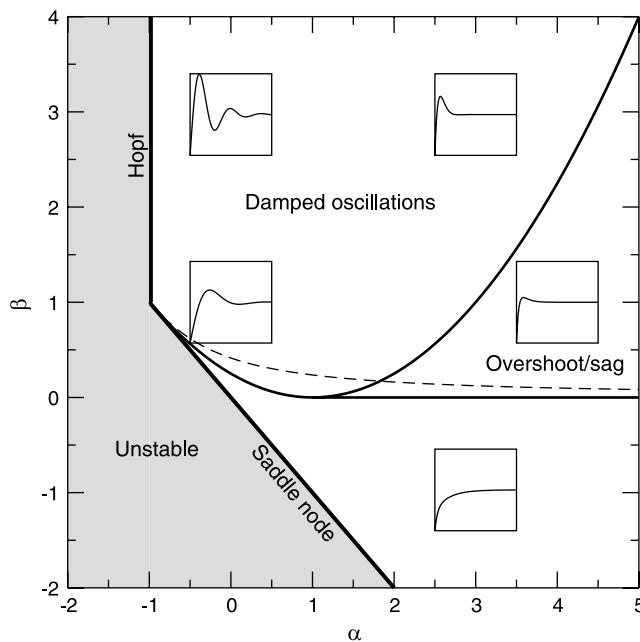


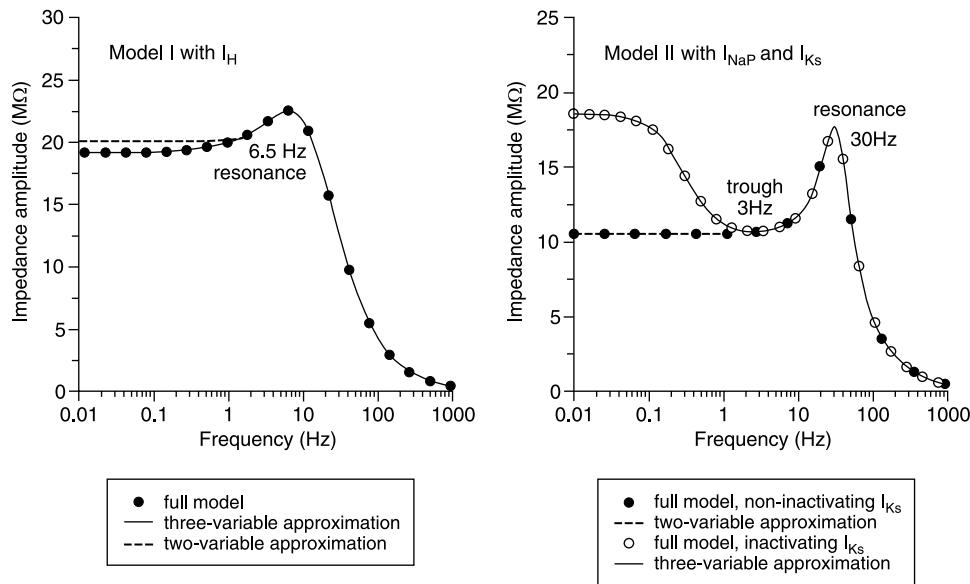
Figure 7.4

Qualitative behaviors of the response to current steps of the GIF model in the $\alpha - \beta$ plane. The gray region again shows the unstable region. Three qualitatively distinct behaviors are possible in the stable region: damped oscillations, overshoot/sag, and simple relaxation. Again, the insets represent voltage responses for representative points on the plane. The dashed line indicates the boundary of the region where subthreshold resonance is found (compare with figure 7.3). (Adapted from Richardson et al., 2003.)

sag if the current pulse is hyperpolarizing); in the remaining stable region in the $\alpha - \beta$ plane, the voltage relaxes exponentially to a new equilibrium value (similar to the LIF model). These regions are shown in figure 7.4. It is interesting that subthreshold resonance and damped oscillations occur in overlapping but not identical regions; there is a parameter region with subthreshold resonance but no damped oscillations, and vice versa.

Reductions of Specific HH-Type Models

Several studies have shown how specific HH-type models can be reduced to two-variable models. This strategy has proven useful for understanding the behavior of several classes of neurons (see Hutcheon and Yarom, 2000 for a review). In particular, these studies have pointed out two mechanisms for generating subthreshold resonance, one involving I_h and another one involving a combination of a slow potassium current and persistent sodium current to amplify the resonance. Reductions of such models to a two-variable model are useful for understanding in which

**Figure 7.5**

Comparison between the impedance of HH-type models and their corresponding reduced two- (or three-) variable GIF models. Both models have standard Na and K Hodgkin-Huxley currents. (a) In addition, model I has an H-current taken from Spain et al. (1987). (b) Model II has a persistent sodium current and a slow potassium current (X.-J. Wang 1993). (From Richardson et al., 2003.)

voltage range qualitatively different behaviors are observed. Comparisons of two conductance-based models studied in Richardson et al. (2003)—a model with an H-current (Spain et al., 1987), and a model with a persistent sodium current and a slow potassium current (X.-J. Wang, 1993)—are shown in figure 7.5 with the corresponding GIF reductions.

Behavior of Models with More than Two Variables

Two-variable models can qualitatively capture impedance curves where there is a single local maximum and no local minimum for $f > 0$ in the impedance curve. For more complicated impedance curves, one has to add additional variables. In fact, one can show that to reproduce the behavior of an impedance curve with n local minima and n ($n + 1$) local maxima, one needs a model with $2n + 1$ ($2n + 2$) variables. For example, if the impedance curve has one local minimum for $f > 0$ followed by a local maximum, a three-variable model is needed (Richardson et al., 2003) (see figure 7.5B).

GIF: Simple Spiking Neuron Model with Subthreshold Resonance

To investigate the effect of subthreshold resonance on spiking dynamics, one can add to the two-variable model defined in equations (7.7) and (7.8) a threshold for spike generation and a reset as in the LIF neuron. Such a model has been termed a “generalized integrate-and-fire” neuron by Richardson et al. (2003). A closely related model is the “resonate-and-fire” model introduced by Izhikevich (2001), which is described by a single complex variable:

$$\dot{z} = (b + i\omega)z + I'_{\text{syn}}(t), \quad (7.14)$$

where $b < 0$ and $\omega > 0$. Note that this model is equivalent to a GIF model with $\alpha = 1$ and $\beta = \omega^2/b^2$. Hence it does not exhibit the variety of behaviors of the GIF neuron; for example, this model always responds to current steps with damped oscillations.

These simplified models have been useful in understanding the implications of subthreshold resonance or damped oscillations on firing behavior. Contrary to LIF neurons, these models can have a type II behavior: a frequency-intensity (f-I) curve in response to constant inputs that is discontinuous at threshold (Brunel et al., 2003), with a region of input currents in which the model is bistable (either silent or firing with some finite frequency). This happens in the regions in which the model responds with damped oscillations (see figure 1.2d) or an overshoot in response to current steps. The minimal firing frequency is then essentially determined by the frequency of the damped oscillations or the time of the overshoot, respectively.

These models have other interesting properties. Models with damped oscillations typically respond optimally when the periods of periodic input spike trains match the frequency of the damped oscillations (Izhikevich, 2001); unlike integrate-and-fire neurons, they fire spikes in response to well-timed inhibitory stimulations (Izhikevich, 2001; Dodla et al., 2006). Finally, the firing rate response of such neurons can have a resonance at either the subthreshold resonant frequency (for sufficiently high noise) or the firing frequency itself (for low noise) (Richardson et al., 2003; Brunel et al., 2003).

Excitability: FitzHugh-Nagumo, Morris-Lecar

To understand the dynamics leading to spike generation, theorists have used two-variable nonlinear models that are idealizations of the Hodgkin-Huxley model. The main advantage of going from the four (or more) variables of Hodgkin-Huxley-type models to two is that it is possible to draw the dynamics on the phase plane of the two variables (see chapter 1, section 3 for phase-plane analysis). The two most popular models in this class are the FitzHugh-Nagumo model (FitzHugh, 1961; Nagumo et al., 1962) and the Morris-Lecar model (Morris and Lecar, 1981). Here, these mod-

els are described only briefly because there are a number of good reviews about phase-plane analysis of such models (e.g., Rinzel and Ermentrout, 1998; Izhikevich, 2006).

The FitzHugh-Nagumo model is defined by the equations:

$$\dot{V} = V - \frac{V^3}{3} - W + I \quad (7.15)$$

$$\dot{W} = 0.08 \times (V + 0.7 - 0.8W), \quad (7.16)$$

where V is the membrane potential, W is a recovery variable, and I is the input current. This model generates spikes when the input current is large enough, owing to the positive feedback in the voltage equation. The repolarization after the spike occurs thanks to the recovery variable W .

The Morris-Lecar model has a more “biophysical” appearance:

$$c \frac{dV}{dt} = -g_{\text{Ca}} m_\infty(V)(V - E_{\text{Ca}}) - g_{\text{K}} w(V - E_{\text{K}}) - g_{\text{L}}(V - E_{\text{L}}) \quad (7.17)$$

$$\tau_w(V) \frac{dw}{dt} = w_\infty(V) - w, \quad (7.18)$$

where V is the voltage; c the capacitance; $g_{\text{Ca}, \text{K}, \text{L}}$ ($E_{\text{Ca}, \text{K}, \text{L}}$) are the maximal conductances (reversal potentials) of the calcium, potassium, and leak currents; $m_\infty(V)$ is a sigmoidal function describing the (instantaneous) activation of the calcium current; and w is the activation of the potassium current, which is described by a first-order differential equation governed by the voltage-dependent time constant $\tau_w(V)$ and the steady-state sigmoidal activation function $w_\infty(V)$. The V and w variables play qualitatively the same roles as in the FitzHugh-Nagumo model. Here, spike generation is due to positive feedback induced by the calcium current, while repolarization is again achieved thanks to the w variable.

These models have been studied extensively using phase-plane analysis (see chapter 1, section 3; Rinzel and Ermentrout, 1998; Izhikevich, 2006) which consists in analyzing the trajectories in the $v - w$ plane. Such an analysis allows one to understand the conditions for the emergence of various types of phenomena: different types of bifurcations leading to firing, corresponding to type I (models that can fire at an arbitrarily small frequency close to threshold), or type II models (a model that cannot fire below a minimal firing frequency), but also excitation block, rebound spikes, and subthreshold oscillations (Makarov et al., 2001), etc.

Spike Generation Dynamics: NLIF Models

If one is interested in the dynamics leading to spike generation but not the repolarization that follows, it is possible to use even simpler models, e.g., a model with a

single voltage variable, with a nonlinearity leading to blowup of the voltage beyond some threshold and a reset of the voltage when it reaches infinity.

Such models can be described by the equation:

$$c \frac{dV}{dt} = -g_L(V - E_L) + \psi(V) + I_{\text{syn}}(t), \quad (7.19)$$

where $\psi(V)$ is a nonlinear function of voltage (Fourcaud-Trocmé et al., 2003). With a suitable choice of $\psi(V)$, the membrane potential diverges to infinity in finite time if the synaptic inputs exceed some threshold. The time of divergence to infinity defines spike timing (it is also possible for practical purposes to define a large but finite voltage threshold). Then, as in the LIF model, the voltage is reset to some potential V_r , after a refractory period τ_{rp} .

The first model that was studied in this class of models was proposed by Ermentrout and Kopell (1986). It is the quadratic integrate-and-fire (QIF) neuron, also known as the theta neuron.¹ In terms of voltage, the dynamics of the model can be written as

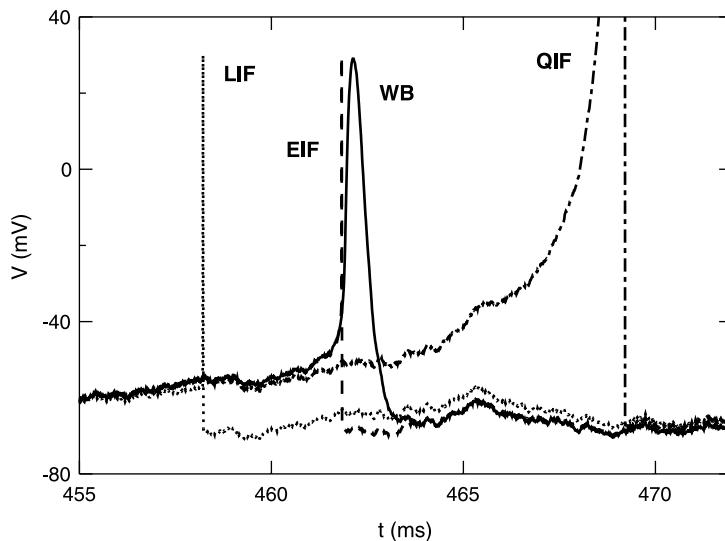
$$c \frac{dV}{dt} = -g_L(V - E_L) + \psi(V) + I_{\text{syn}}(t) \quad (7.20)$$

$$\psi(V) = \frac{g_L}{2\Delta_T}(V - V_T)^2 + g_L(V - E_L) - I_T, \quad (7.21)$$

where V_T is the voltage threshold (the largest stable membrane potential upon injection of a constant current), I_T is the current threshold, and Δ_T is a parameter inversely related to the strength of the nonlinearity, which has units of millivolts.

Using the change in variable $V \rightarrow \tanh(\theta)$, one obtains the theta neuron (Ermentrout and Kopell, 1986; Ermentrout, 1996; Gutkin and Ermentrout, 1998). Such a model (as well as all models described by equation 7.19) can be shown to be type I; the firing rate increases as $(I - I_T)^{1/2}$ when the current is just above threshold, provided the reset is below the voltage threshold. Hence, these simple models can quantitatively capture the f - I curve of any type I neuron model close to the firing threshold, provided the parameters are well chosen.

Whereas the f - I curve depends only mildly on the function ψ , the spike generation dynamics on the millisecond time scale depend markedly on the choice of the function ψ , as shown in figure 7.6. In particular, the quadratic neuron is too slow to accurately capture at the millisecond time scale the spike generation dynamics of models with Hodgkin-Huxley-type fast sodium currents (Fourcaud-Trocmé et al., 2003). For this reason, Fourcaud-Trocmé et al. (2003) introduced the exponential integrate-and-fire model, in which

**Figure 7.6**

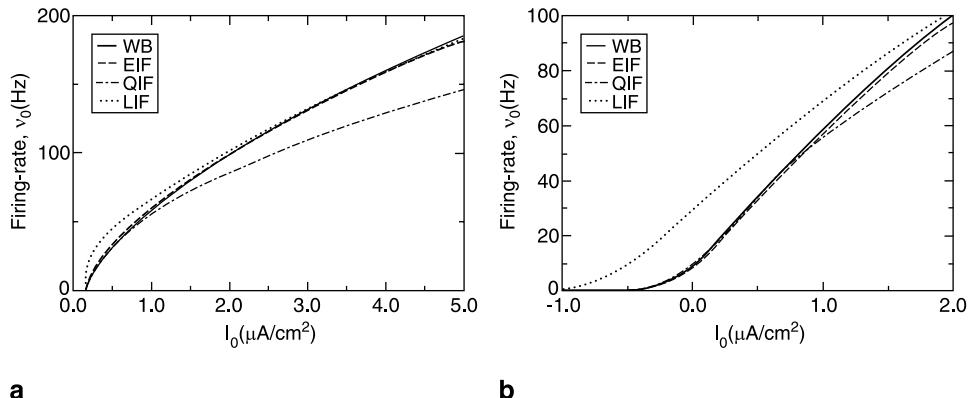
Comparison between the spikes of the Wang and Buzsáki (2006) (WB) model, the LIF model, and two nonlinear IF models: the EIF and the QIF. Only the EIF model captures the spike timing of the WB model at the millisecond time scale. The WB model is a Hodgkin-Huxley model with instantaneous sodium activation and modified parameters to render the model type I. (From Fourcaud-Trocmé et al., 2003.)

$$\psi(V) = g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right). \quad (7.22)$$

The EIF model captures much more accurately the dynamics leading to spike generation than both the LIF (which is too fast) and the QIF (which is too slow), as shown in figure 7.6. Fast spike generation can also be accounted for by a generalization of the theta model introduced by Naundorf et al. (2003, 2005).

Response of NLIF Models to Various Types of Inputs

The fact that NLIF models are described by a single variable makes them amenable to various analytical treatments. In response to constant currents, the firing rate is an increasing function of I , which behaves as $(I - I_T)^{1/2}$ close to threshold. It is also possible to compute the firing rate of QIF and EIF models in the presence of white noise (Lindner et al., 2003; Brunel and Latham, 2003; Fourcaud-Trocmé et al., 2003) and colored noise in the small and long correlation time constant limits (Brunel and Latham, 2003). The f - I curves of EIF, QIF, LIF, and Wang-Buzsáki (WB) models are shown in figure 7.7, in both the absence and the presence of noise.

**Figure 7.7**

Comparison between f - I curves of four models: WB, EIF, QIF, and LIF. (a) In the absence of noise. (b) In the presence of noise. Note that the WB and EIF models are essentially indistinguishable. (From Fourcaud-Trocme et al., 2003.)

Response to Constant Currents

It is also possible to compute the firing rate linear response of the QIF model (Naundorf et al., 2003, 2005) and that of the EIF model in both low and high input frequency limits (Fourcaud-Trocme et al., 2003). The linear firing rate response (susceptibility) describes how the instantaneous firing rate is modulated by an oscillatory input at frequency f :

$$I_{\text{syn}}(t) = I_0 + I_1 \cos(2\pi ft) + I_{\text{noise}}(t) \quad (7.23)$$

$$v(t) = v_0 + v_1(f) \cos[2\pi ft - \phi(f)]. \quad (7.24)$$

In the high frequency limit, $f \rightarrow \infty$, one can show that for QIF and EIF neurons the firing rate modulation decays as a function of frequency as

$$v_1(f) \sim 1/f^\alpha \quad (7.25)$$

$$\phi(f) \sim \alpha\pi/2, \quad (7.26)$$

where $v_1(f)$ is the amplitude of the firing rate modulation; $\phi(f)$ is its phase relative to the input current; and α is an exponent, which is $\alpha = 1$ for the EIF neuron and $\alpha = 2$ for the QIF neuron. This exponent is independent of whether noise is white or colored. This behavior is drastically different from the LIF neuron, whose exponent is $\alpha = 0.5$ in the presence of white noise and $\alpha = 0$ in the presence of colored noise. Remarkably, the susceptibility of the EIF model is also essentially indistinguishable

from that of the WB neuron, which also has $1/f$ attenuation at high f . This $1/f$ behavior can be understood from the fact that the activation curve of a fast sodium current can be closely approximated by an exponential close to the firing threshold. Hence, if one neglects the very fast time constant of the activation variable, one obtains a fast sodium current that has a functional form that is very similar to an exponential close to the firing threshold. The phase shift in the HH-type model is also consistent with EIF with the addition of a small delay of ~ 0.2 ms to account for spike time-to-peak.

IF with Firing Rate Adaptation

Currents leading to spike frequency adaptation typically increase transiently during spikes and then decay between spikes. These currents tend to hyperpolarize the membrane potential, thereby providing a negative feedback of the firing frequency of the neuron onto itself.

This type of current can be captured qualitatively by a single variable that makes an instantaneous jump whenever the neuron emits a spike and decays exponentially between spikes (Treves, 1993; Ermentrout, 1998; Ermentrout et al., 2001; van Vreeswijk and Hansel, 2001; Fuhrmann et al., 2002; Benda and Herz, 2003; Meunier and Borejsza, 2005):

$$c \frac{dV}{dt} = -g_L(V - E_L) - I_a + I_{\text{syn}}(t) \quad (7.27)$$

$$\tau_a \frac{dI_a}{dt} = -I_a. \quad (7.28)$$

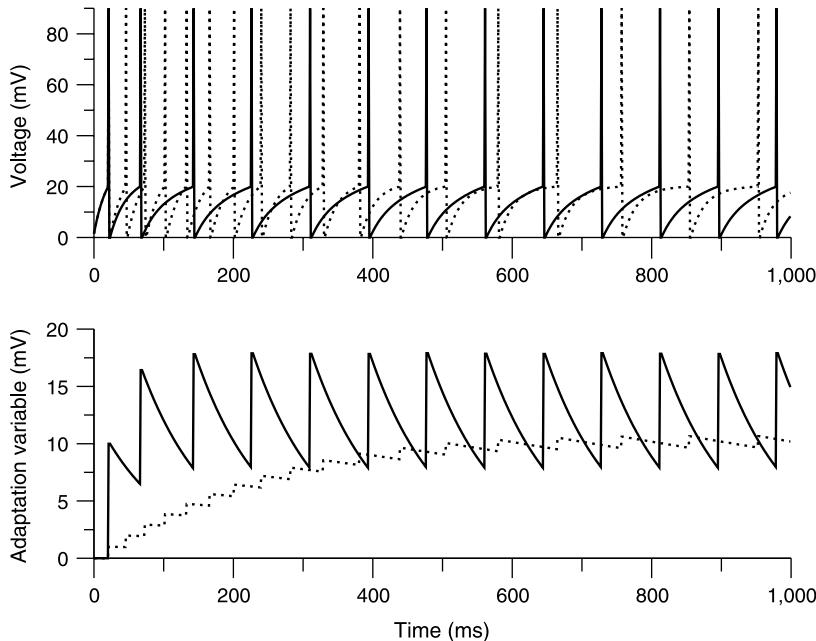
When the membrane potential reaches the firing threshold $V = V_T$, the voltage is reset to V_R and the adaptation current I_a makes a jump of size d :

$$V \rightarrow V_R \quad (7.29)$$

$$I_a \rightarrow I_a + d. \quad (7.30)$$

The dynamics of such a model is shown in figure 7.8. Note that a variant of this model incorporates a nonlinearity in the dynamics of the adaptation variable to account for a saturation of the adaptation current at high frequencies (Meunier and Borejsza, 2005).

This adaptation variable has several important effects at the single neuron level. First, it tends to linearize the f - I curve of the neuron (Ermentrout, 1998). Second, it tends to decrease the gain of the firing rate in response to low-frequency inputs and induces a phase advance of the firing rate in response to such inputs. This in turn can lead to oscillations in networks of excitatory neurons (Fuhrmann et al., 2002).

**Figure 7.8**

Dynamics of an integrate-and-fire neuron with adaptation in response to a current step. Neuron initially at $V = I_a = 0$. Top: membrane potential. Bottom: Adaptation variable. Solid line: $\tau_a = 100$ ms, $d = 10$ mV. Dotted line: $\tau_a = 1$ s, $d = 1$ mV.

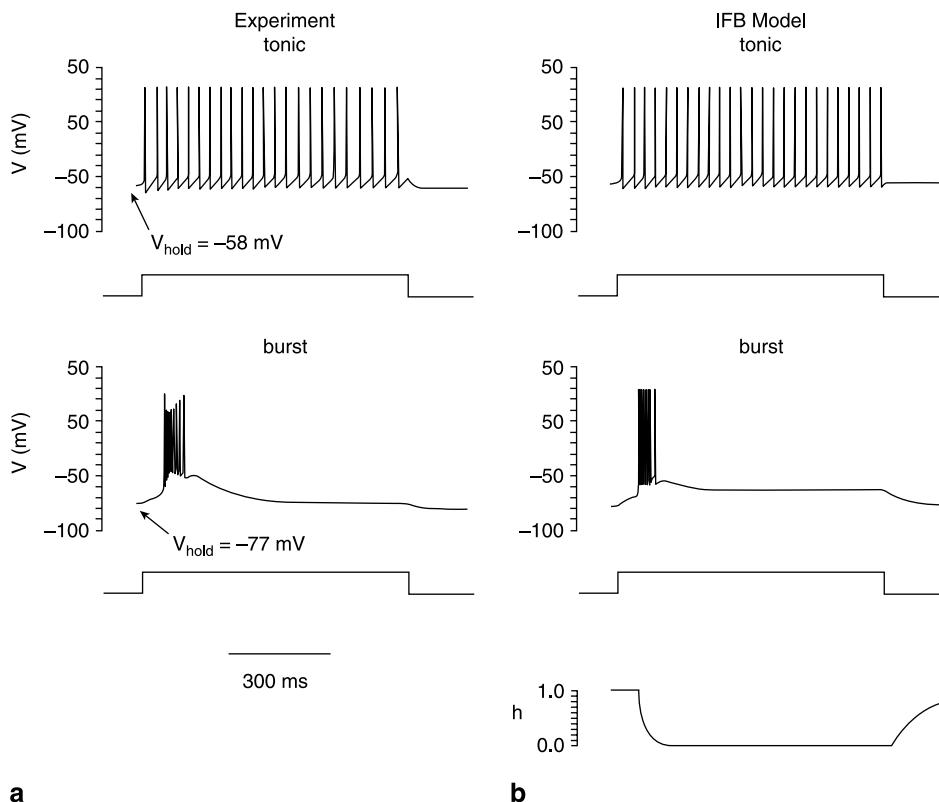
Bursting: Integrate-and-Fire-or-Burst

In order to reproduce the basic features of the bursting behavior of thalamocortical relay neurons, Smith et al. (2000) introduced an integrate-and-fire-or-burst model by adding an additional slow variable to the basic integrate-and-fire model:

$$c \frac{dV}{dt} = -g_L(V - E_L) - g_T \Theta(V - V_h)h(V - E_T) + I_{\text{syn}}(t) \quad (7.31)$$

$$\frac{dh}{dt} = \begin{cases} -h/\tau_h^- & (V > V_h) \\ (1-h)/\tau_h^+ & (V < V_h), \end{cases} \quad (7.32)$$

where the Heaviside function $\Theta(V - V_h)$ represents an instantaneous idealization of the fast activation of the low-threshold calcium current I_T , while the slow variable h represents the inactivation of the low-threshold calcium. The dynamics of h is governed by equation (7.32): it relaxes to zero with time constant τ_h^- when the voltage is above V_h , while it relaxes to unity with a slower time constant τ_h^+ when the voltage is below V_h .

**Figure 7.9**

Integrate-and-fire-or-burst (IFB) versus thalamic relay cell. Both real and model neurons emit a periodic train of single spikes when they are initially at a depolarized membrane potential (I_T inactivated). They both emit a single burst when initially at a hyperpolarized membrane potential, owing to activation of I_T followed by inactivation. (From Smith et al., 2000.)

Smith et al. (2000) have shown that this model captures the qualitative features of the bursting dynamics of thalamic relay cells, as shown by figure 7.9. In particular, when a current step is applied to the model, the response depends drastically on the initial membrane potential: (1) If V is initially below V_h , the model generates a single burst of spikes. This is because initially $h = 1$; when the voltage is depolarized to a value larger than V_h , the I_T current provides a strong depolarization that generates the burst. The burst is terminated because of the decay of the variable h to zero with a time constant τ_h^- . (2) If V is initially above V_h , then $h = 0$; hence the I_T current remains inactivated and the dynamics is identical to that of an LIF neuron.

Izhikevich and Related Two-Variable Models

All the models discussed so far include one or both of the following features: a nonlinearity in the equation for the voltage dynamics to capture spike generation dynamics; and/or a second variable that allows the model to reproduce such features as subthreshold resonance, firing adaptation, or bursting. This led Izhikevich (2003) to propose a minimal two-variable model that can be set to perform any of the these behaviors, provided its parameters are tuned accordingly. The model has two variables, v (voltage) and u (adaptation), which evolve, for $v < v_{\text{peak}}$ according to Izhikevich (2003, 2004, 2006):

$$c \frac{dv}{dt} = k(v - v_r)(v - v_t) - u + I_{\text{syn}}(t) \quad (7.33)$$

$$\frac{du}{dt} = a[b(v - v_r) - u], \quad (7.34)$$

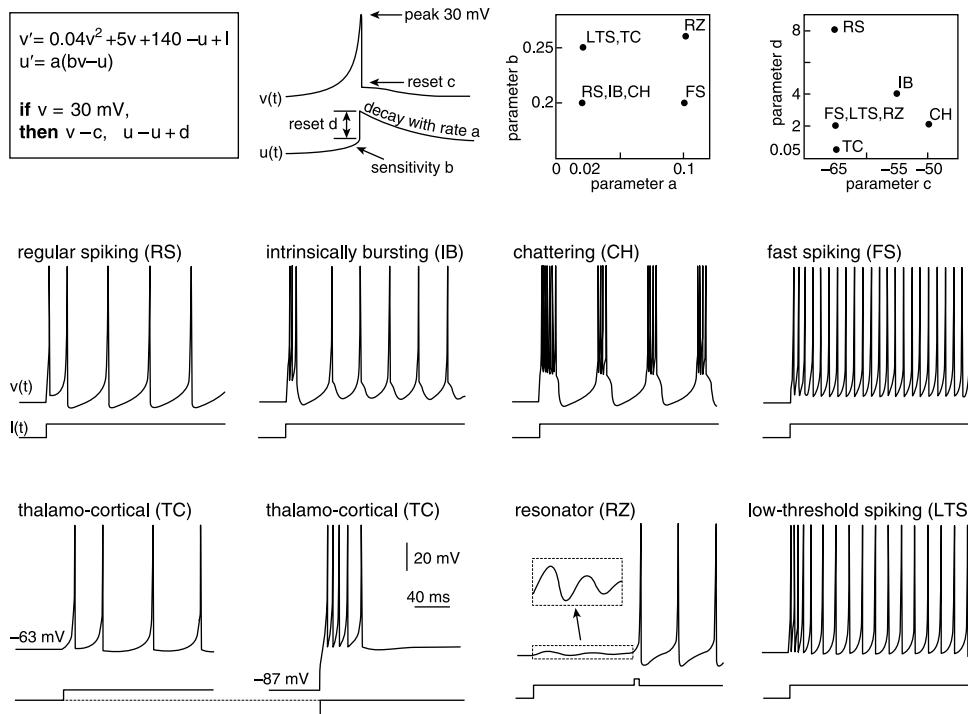
where a is the inverse of the time constant of the u dynamics, b measures how strongly the variable u is coupled to the voltage, v_r is the membrane resting potential, and v_t is the instantaneous threshold potential. If $v = v_{\text{peak}}$, a spike is emitted and both variables u and v are reset according to

$$v \rightarrow c' \quad (7.35)$$

$$u \rightarrow u + d. \quad (7.36)$$

Note that for $b = d = 0$ the model reduces to the QIF neuron with a finite spike threshold. For $b = 0$, $d > 0$ the variable u is a pure adaptation variable as in the LIF model with adaptation. On the other hand, for $b > 0$, $d = 0$ the variable u is similar to the second variable in the GIF model, potentially giving rise to subthreshold resonance. Finally, the variable c' can be set to regulate the “burstiness” of the model: for a large c' , the model will tend to emit a spike immediately after a first spike; the burst will then be terminated by the buildup of variable u . In fact, Izhikevich (2003, 2004) showed that depending on the values of a , b , c' , and d , the model can have a wide variety of firing patterns (see figure 7.10).

We argued earlier that a model with a quadratic nonlinearity is actually not the best candidate for obtaining spike generation dynamics. In fact, an exponential nonlinearity seems more suited for capturing spike generation at the millisecond time scale. This is what led Brette and Gerstner (2005) to propose the so-called adaptive EIF, which is essentially an EIF model with an “adaptation variable” described by an equation of the type of equation (7.34), or equivalently, an Izhikevich model with the quadratic nonlinearity replaced by an exponential one. Brette and Gerstner

**Figure 7.10**

Diversity of behaviors of the Izhikevich model. (From Izhikevich, 2003. See Izhikevich, 2006 for more details.)

(2005) have demonstrated that such a model can reproduce the full spike train of an HH-type model of a cortical pyramidal cell (McCormick et al., 1993) under noisy current injection with about 95% accuracy. The dynamics of the adaptive EIF model have been analyzed by Touboul and Brette (2008) and by Naud et al. (2008).

7.5 Conclusions

The models that have been described in this chapter are minimal models that have proven extremely useful for understanding some basic features of single-neuron dynamics, for example, irregularity of firing, subthreshold resonance, bursting, spike generation dynamics, the effect of adaptation on firing. Furthermore, such models are also extremely helpful at the network level; their level of simplification often allows theorists to analyze network dynamics mathematically (see, e.g., Abbott and van Vreeswijk, 1993 and Brunel and Hakim, 1999 for examples of such analysis). They are also often used in large network simulations; see, e.g., Brette et al. (2007)

for a review of various software tools and which models can be simulated by each tool.

One question of interest for experimentalists or theorists is how to choose a model and the parameters of that model. The answer of course depends on the nature of the problem one is interested in. Theorists have often used the LIF neuron as a convenient tool for analyzing network dynamics. However, one should be aware of the pathologies of such a model. In the absence of noise, its f - I curve is nongeneric. In the presence of noise, its behavior in response to fast-fluctuating inputs is also non-generic. Furthermore, both pathologies are present in all threshold models, such as the GIF, AIF, and IFB. However, if one is not interested in the dynamics at the millisecond time scale, but only in phenomena occurring at a slower time scale, then the LIF model and its threshold variants might still be a reasonable choice. In fact, Rauch et al. (2003) have shown that the f - I curve of cortical pyramidal cells can be well fitted by LIF neurons with an adaptation variable.

If capturing the dynamics at the millisecond time scale is crucial, then a model with more realistic spike generation dynamics should be used. We have argued here that an exponential nonlinearity seems the best way to capture the spike generation of Hodgkin-Huxley-type models. However, it is still an open issue whether spike generation in cortical pyramidal cells is actually well described by a Hodgkin-Huxley model (Naundorf et al., 2006; McCormick et al., 2007). In fact, spike generation in cortical cells seems much sharper than in one-compartment models with standard Hodgkin-Huxley sodium currents. This could be well described by an EIF model with small Δ_T , which would be effectively in-between the LIF and the HH model. A recent study shows that the instantaneous I - V curve of cortical pyramidal cells can be well fitted by an EIF with $\Delta_T \sim 1$ mV (Badel et al., 2008).

The LIF or EIF models in their most basic form do not incorporate variables that give rise to dynamics at a slower time scale than that of spike generation, such as subthreshold resonance, firing rate adaptation, or bursting. To capture such dynamics, a second variable has to be coupled to the voltage variable. If the goal is to model a specific cell type, parameters of such an additional variable can then be fitted to experimental data. In the case of a cell type showing subthreshold resonance, parameters of the second variable can be fitted using the impedance curve. In the case of firing rate adaptation, parameters can be fitted using the f - I curve. Finally, parameters for a bursting neuron can be constrained by the duration of the burst as well as voltage ranges from where bursts can be elicited.

To summarize, simplified models that are in-between the LIF neuron and HH-type models have proven very helpful in understanding the wide spectrum of dynamic properties observed in real neurons. Furthermore, in some instances such minimal models can be shown to reproduce the behavior of real neurons, not only qualitatively, but even quantitatively. Finally, most (if not all) of our understanding of the

dynamics of networks of spiking neurons has come from studies that use such simplified model neurons. For these reasons, integrate-and-fire-type neurons have become indispensable tools for theoretical neuroscientists.

Acknowledgments

I would like to thank the people with whom I have had the pleasure of discussions or an exchange of correspondence on issues related to the topics in this chapter: Larry Abbott, Frances Chance, Nicolas Fourcaud-Trocmé, Stefano Fusi, Caroline Geisler, Michele Giuliano, Vincent Hakim, David Hansel, Bruce Knight, Giancarlo La Camera, Peter Latham, Magnus Richardson, Walter Senn, Simone Sergi, Mark van Rossum, Carl van Vreeswijk, Xiao-Jing Wang. Thanks also to Eugene Izhikevich and Claude Meunier for comments on earlier versions of the manuscript.

Note

1. In fact, Alan Hodgkin had already introduced this model in the 1940s to fit $f\text{-}I$ curves of leg nerve fibers of the crab *Carcinus*, and the model was studied subsequently by Bruce Knight (Bruce Knight, personal communication).

8 Reconstruction of Neuronal Morphology

Gwen Jacobs, Brenda Claiborne, and Kristen Harris

The goal of this chapter is to describe methods for obtaining high-quality, accurate 3-D reconstructions of neuronal morphology for use in constructing computational models (chapters 10 and 11). Because the accuracy of any model depends on the structural characteristics of the neuron as well as its biophysical properties, capturing the lengths and diameters of neuronal processes and the locations of synapses is critically important for developing realistic models. There are numerous sources of error that can be introduced during histological processing and tissue sectioning, as well as during the processes of neuronal imaging and digitizing, that can affect the quality of the model. This chapter summarizes the techniques currently in use for obtaining morphological parameters of neurons at both the cellular and ultrastructural levels and provides suggestions for reducing the errors introduced during the process. A description and evaluation of reconstruction techniques that employ both open-source and commercial software programs for obtaining 3-D reconstructions are included. There is now a growing number of accessible archives of 3-D reconstructions of neuronal morphology for use by the neuroscience and modeling communities.

8.1 Neuron Staining Techniques and Tissue Preparation for Light Microscopy

Whereas precise measurements of synapse distributions require the use of electron microscopy, accurate reconstructions of neuronal branches as well the relative distributions of dendritic spines and axonal boutons can be obtained using light microscopy (see Shepherd and Harris, 1998 for discussion). Any one of a number of different labeling methodologies can be used to visualize neuronal structure at the light microscope level. Each technique has its advantages and disadvantages and thus the choice of a labeling method will depend on several factors, including the goal of the experiment, the source of the tissue (such as vertebrate or invertebrate), and the region of the nervous system to be examined. Labels that can be injected into the cell are preferred for modeling studies because the electrode that is used for dye injection can also be used to obtain electrophysiological data on the same

neuron. Here we begin our discussion by focusing on those labels that are most commonly employed for intracellular injections, and then we consider other staining techniques that also provide excellent visualization of neuronal structure at the light microscope level.

Staining Neurons by Intracellular Injection

In selecting a label, an investigator must first consider whether a dense product or a fluorescent label is most suitable. Compounds that produce dense products are most often chosen for the analysis of dendritic trees and axons at the light microscope level (because they do not undergo photobleaching) and for studies that require combined light and electron microscope imaging. Early investigators used horseradish peroxidase (HRP) to label vertebrate neurons either *in vivo* or *in vitro* (Claiborne et al., 1986, 1990; Ishizuka et al., 1995) and cobalt chloride to label invertebrate neurons (Bacon and Altman, 1977; Levine and Murphey, 1980; S. E. Johnson and Murphey, 1985). Currently, biocytin is the most common intracellular label for both vertebrate and invertebrate neurons when a dense product is required (Horikawa and Armstrong, 1988; McDonald, 1992; Heinrich et al., 1998; Pyapali et al., 1998; X. J. Sun et al. 1998; Jaeger, 2001).

Fluorescent dyes are preferred for studies that require visualization of the neuron during the labeling procedure or that require the use of a laser-scanning microscope to analyze both neuronal branching patterns and the details of other morphological features such as dendritic spines or axonal boutons. One of the first available dyes for intracellular labeling was Lucifer Yellow (W. W. Stewart, 1978). Although it is still used for quantitative studies of neurons in fixed slices (Duan et al., 2002, 2003), it is less than ideal for labeling studies. It is inefficiently excited with an argon-ion laser and undergoes relatively rapid photobleaching. More recently, fluorescent hydrazide and hydroxylamine derivatives have been developed for intracellular staining (Alexa dyes; Invitrogen Molecular Probes); they are photostable, they can be fixed with aldehyde-based fixatives, and they are water soluble.

Staining Vertebrate Neurons with Biocytin

Biocytin can be used for intracellular labeling either *in vivo* or *in vitro* (Pyapali et al., 1998; Chitwood et al., 1999; Staiger et al., 2004; for a detailed review, see Jaeger, 2001). Here we focus on the use of biocytin in brain slices.

To prepare tissue slices for *in vitro* work, the animal is deeply anesthetized and the brain removed. The region of interest can be quickly dissected away from the rest of the tissue or the brain can be left intact. If preferred, the anesthetized animal can be perfused first through the heart with ice-cold saline to remove erythrocytes; blood cells contain endogenous peroxidase and will stain with the procedures required to visualize the biocytin (or horseradish peroxidase; Claiborne et al., 1986). To optimize

viability, the tissue should be sectioned immediately; a vibratome is most commonly used. Slice thickness will depend on the type of neuron to be labeled, but most investigators cut sections that are between 300 and 400 μm . If the slices are much thinner, one risks severing many dendrites on most cortical cells (see chapter 9), and if they are thicker, the viability of tissue toward the center of the slice is compromised, owing to hypoxia. To allow dendritic spines to stabilize, slices should be allowed to recover for 3 hr prior to beginning the experiment (Kirov et al., 2004).

Recordings and injections can be made using either sharp electrodes or patch electrodes. Sharp electrodes filled with 2–4% biocytin (Sigma) or Neurobiotin (Vector Laboratories) by weight, dissolved in 1 M potassium acetate, have good recording characteristics, and either compound can be injected using positive current pulses of 100–200 ms duration and an amplitude of 1–4 nA applied for 5–10 min at 1–4 Hz after electrophysiological data are obtained (Pyapali et al., 1998; Jaeger, 2001). Slices should remain in the chamber for 15 to 60 min (depending on the size and complexity of the neuron) after the injection is complete to allow the label to diffuse throughout the dendritic tree. When patch pipettes are used in the whole-cell configuration, concentrations of biocytin or Neurobiotin between 0.1 and 0.5% are normally employed and, in most cases, the neuron will fill by passive diffusion over the course of 30 to 60 min (Chitwood et al., 1999; Wang et al., 2002; Staiger et al., 2004; Golding et al., 2005).

To ensure the probability of staining neurons with complete dendritic trees, neurons with somata located near the center of the slice should be selected (Claiborne et al., 1990; Rihm and Claiborne, 1990; Mainen et al., 1996; Carnevale et al., 1997). This is often difficult to accomplish when patch recordings are done under visual control because only cell bodies relatively close to the surface are easily imaged with infrared differential interference microscopy. Thus investigators must be extra vigilant in noting severed dendrites during the reconstruction process, as discussed in more detail later and in chapter 9. Two other caveats are important. Slices should not be allowed to remain in the chamber for more than the minimal time required for label diffusion. If a neuron begins to die before it is fixed, swellings can occur so that a labeled process looks like a string of beads (see figure 6.1 in Jaeger, 2001). These artifacts in some adult neurons, however, should not be confused with the normal dendritic varicosities that often characterize interneurons or neurons from young animals (Jones et al., 2003). It is worth noting that chilling brain slices can also cause beading that disappears once the slices are warmed (Kirov et al., 2004). In addition, after recordings and intracellular injections are completed, the electrode should be removed carefully so that the injected neuron sustains minimal damage to the cell body and proximal dendrites.

After label injection, slices are fixed in 4% paraformaldehyde and 1.25% glutaraldehyde overnight, washed in buffer, and sectioned at 100 μm with a vibratome. Prior

to reaction with avidin coupled to horseradish peroxidase, slices can be made more permeable with a quick-freezing protocol (Jaeger, 2001) and incubated in 0.5% H₂O₂ for 30 min to inactive endogenous peroxidases. Slices are then incubated in avidin-horseradish peroxidase (Vectastain ABCkit, Vector Laboratories, Inc., Burlingham, CA) for either 3 hr at room temperature, or overnight at 4°C, for best results. Next, slices are incubated in a solution of 1 ml of 0.05% diaminobenzidine (DAB) and 1 mM NiCl₂ in 0.1 M phosphate buffer for 1 hr on a shaker table. Three to four drops of 3% H₂O₂ in buffer are added to each vial and the slices incubated an additional hour. After washing in buffer, tissue can be mounted in an aqueous mounting medium (Golding et al., 2005) or cleared in ascending concentrations of glycerol and stored and mounted in 95% or 100% glycerol. The use of glycerol for dehydration and clearing minimizes tissue shrinkage and the artifacts that accompany dehydration with alcohol series. If needed, the HRP & DAB product can be intensified by rehydrating the slices in descending concentrations of glycerol, followed by a wash in 50 mM Tris buffer (pH = 7.4). Slices are then incubated in 2 ml of 0.1% cobalt chloride, 0.1% nickel ammonium sulfate, and 0.05% DAB for up to 45 min. The slices should be checked periodically, and when nonspecific tissue darkening begins to occur, washed in buffer and cleared and mounted in glycerol. If long-term storage of stained neurons is desired, an alternative method of hardening the tissue with osmium can be used (Staiger et al., 2004). This method produces minimal shrinkage of the tissue and less fading of the stained neurons.

Staining Invertebrate Neurons with Biocytin

The techniques for intracellular staining of neurons in invertebrates with biocytin are essentially the same as those described here for vertebrate preparations. Intracelluar injections are done *in situ* in semi-intact animals. Best results are obtained when the preparation is perfused with oxygenated saline and the tracheae (air supply) are intact. In most cases, the ganglion containing the stained neuron can be processed as a whole mount using protocols similar to those described earlier; however modifications of the procedure are required to achieve good penetration of the reactants into the tissue. Insect ganglia are covered in a thick glial sheath that requires softening with protease or incubation with detergents (Staudacher, 1998). Invertebrate neurons are often much larger in size than vertebrate neurons, requiring longer injection and diffusion times for optimal staining. Sometimes injection with biocytin results in light or incomplete staining of the extensive dendritic trees, precluding a complete anatomical reconstruction of the cell.

Staining Invertebrate Neurons with Cobalt Chloride

Injection of cobalt chloride is an alternative method used to stain neurons in invertebrate ganglia and results in a very dark stain and complete filling of all dendritic processes. Cobalt chloride can be injected intracellularly with pulses of positive current

(5 nA at 1 Hz) for a brief period of time (1–2 min). Cobalt ions diffuse very quickly through the cell, so long injection or diffusion times are not necessary to stain the cell in its entirety (Levine and Murphey, 1980). Cobalt is highly toxic to the cell, so the tissue should be processed quickly following staining to avoid artifacts such as blebbing and swelling. The ganglion (or brain) containing the stained cell is dissected from the animal and immersed in saline saturated with hydrogen sulfide for 15 min. The tissue is rinsed in buffer and fixed in Carnoy's fixative (60% ethanol (95%), 20% chloroform, and 10% glacial acetic acid) for an hour. The tissue can be stored in 70% ethanol for up to a week before further processing. The tissue is then rehydrated in a descending series of ethanol and incubated in a solution of gum acacia, hydroquinone, and citric acid for an hour at 50 °C, in the dark. Silver nitrate is then added to the solution and the tissue is incubated in the dark until the staining develops. The tissue is dehydrated in ethanol, cleared in methyl salicylate, and mounted in Canada balsam. This method stains cells completely with a black precipitate; however, the tissue shrinks by 50% or more during fixation and dehydration. These preparations can be stored mounted in Canada balsam for years with no fading of the stain. Detailed methods for this procedure can be found in Bacon and Altman (1977) or S. E. Johnson and Murphey (1985).

It should be noted that while cobalt chloride can be successfully injected into invertebrate neurons, it can also be applied extracellularly. For example, cricket mechanosensory afferents can be stained with these same methods by introducing cobalt chloride into the lumen of a cut hair and allowing the cobalt to diffuse into the sensory neuron in the intact animal overnight at 4 °C (S. E. Johnson and Murphey, 1985; Jacobs and Nevin, 1991).

In some cases, injection of Neurobiotin or biocytin into invertebrate neurons results in the staining of multiple cells, via diffusion through gap junctions of electrically coupled cells. Alexa dyes, with their higher molecular weights, can be used as a substitute since they are confined to the injected cell (Fan et al., 2005). The electrodes filled with Alexa have good recording characteristics, although not as good as those filled with biocytin alone.

Alternative Staining Methods for Obtaining Neuronal Morphologies in Vertebrate Tissues

There are four other types of techniques in common use that do not require intracellular dye injection: Golgi impregnation techniques (Desmond and Levy, 1982), dextram amine staining (Megias et al., 2001), lipophilic dye labeling (Rahimi et al., 2003; Garza-Meilandt et al., 2006), and fluorescent labeling through expression of endogenous green fluorescent protein (GFP) in transgenic mice (Feng et al., 2000; Mizrahi and Katz, 2003). These latter methods produce excellent structural data although they do not allow the measurement of physiological parameters.

The Golgi Method

Golgi impregnation in fixed tissue produces random staining of multiple neurons within the tissue. Although this method is less suitable for modeling studies, neurons are stained intensely and completely, albeit in an unpredictable fashion. This technique is useful for anatomical studies requiring tissue that is difficult to obtain, such as human pathology specimens. The high contrast between the stained neuron and surrounding tissue makes the anatomical reconstruction process easier than with other more lower-contrast methods (see K. M. Brown et al., 2005). For detailed protocols of the rapid Golgi technique, see Sultan and Bower (1998) or Desmond and Levy (1982).

Biotinylated Dextran Amine

Another useful label that results in a dense product is biotinylated dextran amine or BDA (Molecular Probes). BDA is injected into the brain region of interest and after a survival period the animal is anesthetized and perfused, the brain is sectioned, and the sections are processed for visualization of the stained neurons (for detailed methods, see Megias et al., 2001). This methodology results in Golgi-like labeling of single neurons, although often large numbers of neurons close to the injection site are labeled. BDA is particularly advantageous for those experiments requiring both light and electron microscopy.

Labeling with Lipophilic Dyes

Lipophilic dyes such as DiI (1,1'-dioctadecyl-3,3',3'-tetramethylindocarbocyanine perchlorate; Molecular Probes) can be used to label cells in live or fixed tissue, producing an intensely fluorescent label that can be used with confocal imaging techniques and subsequent reconstruction from confocal images (described later). Most commonly, DiI labeling is done in fixed tissue. To prepare the tissue for labeling, animals are anesthetized and perfused with cold 4% paraformaldehyde and 0.5% glutaraldehyde in 0.1 M phosphate buffer ($\text{pH} = 7.4$). The brains are removed and left in the cold fixative for 1 hr and then washed in buffer. The fixation time should be closely monitored because longer fixation times lead to less than optimal labeling. Brain slices (200–400 μm) are cut using a vibratome and the thickness of the slice should be chosen to ensure that an entire neuron is contained within the slice. Crystals of DiI are crushed and small crystals are picked up on the tip of a micropipette and placed directly on the slice in proximity to the neurons of interest. For example, to label pyramidal neurons in hippocampal region CA1, crystals are placed at the top of the stratum lacunosum-moleculare, the layer containing the distal dendritic tips (Garza-Meilandt et al., 2006). Labeled slices should be stored in 2% paraformaldehyde for 2–3 days to allow diffusion of the dye throughout the neurons and then mounted in buffer; because DiI is lipophilic, alcohol- and glycerol-based mounting media cannot be used. Labeled neurons can be imaged with a confocal microscope

and the neurons reconstructed from the image stacks as described later (Rahimi et al., 2003). It is worth noting that although Dil labeling is often most impressive in tissue from young animals (Jones et al., 2003), recent studies show that it can produce excellent labeling in tissue from young adult and aged rodents (Kirov and Harris, 1999; Rahimi et al., 2003; Garza-Meilandt et al., 2006).

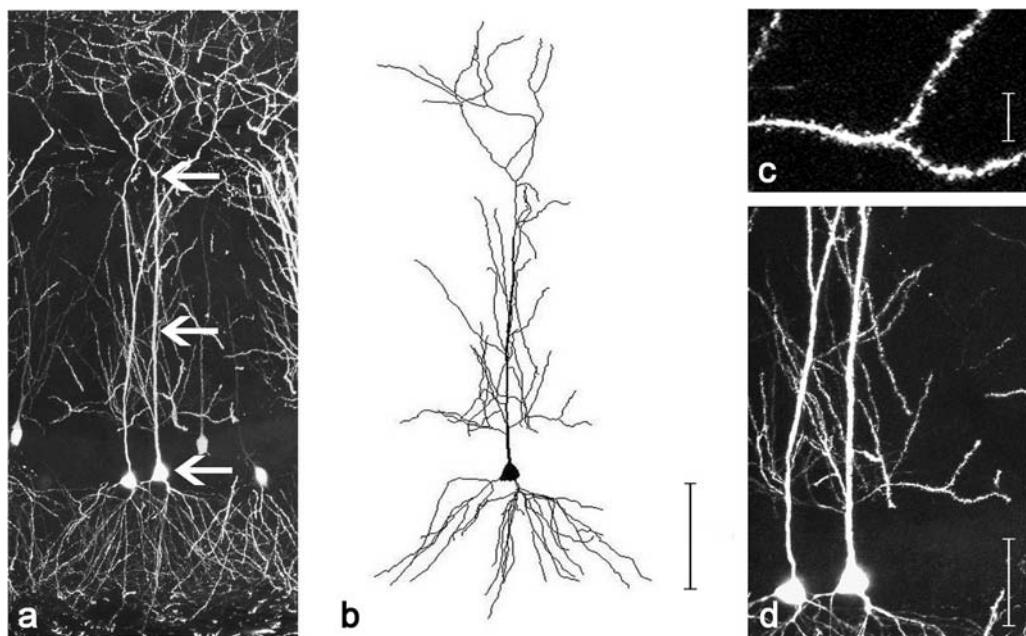
Fluorescent Labeling with Green Fluorescent Protein in Transgenic Mice

Many lines of transgenic mice have been developed that express GFP in specific populations of neurons (Feng et al., 2000; Mizhra and Katz, 2003). Varying numbers of neurons express GFP in the different lines. Two lines are particularly useful for quantifying the morphology of hippocampal neurons: the “M” line and the “O” line (Feng et al., 2000). In these lines, GFP is expressed under the control of neuron-specific elements from the *thy1* gene, an immunoglobulin gene that is expressed by projection neurons in many parts of the nervous system, as well as in some non-neuronal cells. Deletion of a specific intron abolishes expression in non-neuronal cells. Excellent GFP expression is found in 24-month-old mice (Perez et al., 2005).

To prepare the tissue, animals are deeply anesthetized with pentobarbital (130 mg/kg) and perfused with 4% paraformaldehyde in 0.1 M phosphate buffer (pH = 7.4). The brain is removed from the skull, left in the fixative for 3 hr, washed in buffer, and sectioned coronally using a vibratome at 200 μ m. The sections are mounted in Vectashield (Vector Inc., Burlingame, CA). Fluorescently labeled neurons can be examined using a microscope equipped with a fluorescent attachment, the appropriate filter (DiI, Chroma No. 41002; GFP, No. 31001; Chroma Technology Corp., Brattleboro, VT), and a 63 \times Zeiss oil immersion objective (NA 1.25; 0.5-mm working distance). After applying selection criteria, acceptable neurons are imaged with a confocal laser scanning microscope and dendritic trees reconstructed as described later (figure 8.1).

Alternative Staining Methods for Obtaining Neuronal Morphologies in Invertebrate Tissues

GFP has been used extensively for studying neuronal morphology in invertebrates, most commonly in *Caenorhabditis elegans* (Chalfie et al., 1994) and in *Drosophila* (Brand, 1995) either as an endogenously expressed protein or in conjunction with the GAL4 system, which permits *in vivo* labeling of neurons and other cell types throughout development (Brand and Dormand, 1995; Ito et al. 1997). GFP can also be used to label the entire structure of neurons homozygous for specific genes in the intact brain using the mosaic analysis with repressible cell marker (MARCM) technique (Lee and Luo, 1999). Neurons labeled in this way can be reconstructed from stacks of confocal images for morphometric analysis. In experiments in which it is important to view the GFP-labeled neuron in conjunction with the ganglion

**Figure 8.1**

Pyramidal neurons from the CA1 region of the mouse dorsal hippocampus. (a) A stacked series of confocal images of CA1 pyramidal neurons expressing green fluorescent protein in a slice from an adult transgenic male mouse (Feng et al., 2000). (b) Computer-generated reconstruction of the pyramidal neuron indicated by arrows in a and shown at the same scale as a. Scale bar, 100 µm. (c) and (d) Higher magnification images of dendritic trees illustrated in a. (c) Tertiary dendrite with bifurcation at the arrow. The location of this dendrite within the tree is indicated by a corresponding arrow in d. Scale bar, 10 µm. (d) Stacked series of confocal images of the two neurons shown in a. Only a subset of the images used in a is included in the stack here to illustrate that individual dendrites of interest can be traced without interference from other labeled neurons when the number of scans in the stack is reduced. The arrow indicates the bifurcating dendrite shown in c. Scale bar, 50 µm.

architecture, a peroxidase-linked antibody to GFP can be used with silver staining techniques in semithin sections (Tyrer et al., 2000). This method has the advantage of viewing GFP-labeled neurons or processes in context with major anatomical landmarks, such as commissures, axon tracts, and neuropils.

Suggestions for Avoiding Artifacts Introduced during Histological Processing of Tissue

The protocols required for staining single neurons for subsequent reconstruction can easily destroy the value of the tissue itself. Injection of biocytin or cobalt chloride (especially), long recording times, or hypoxia can introduce structural artifacts (including blebbing that may easily be mistaken for synaptic varicosities). Such capricious, nonuniform distortions can never be adequately compensated for during recon-

struction and must be eliminated through refinement of injection and histological protocols.

Dehydration of the stained and fixed tissue through an ascending alcohol series, followed by “clearing” in a reagent such as methyl salicylate or other mounting media inevitably leads to overall shrinkage of the preparation. Shrinkage can be as large as 50% with some standard protocols. Any shrinkage will result in a systematic underestimate of morphological parameters. As described earlier, dehydration in glycerol reduces shrinkage to a minimum but can still result in some artifacts.

A practical approach toward monitoring and compensating for any residual artifacts that cannot be eliminated through an optimization of histological protocols is as follows: (1) Use a fluorescent marker in conjunction with the dye used for staining. (2) After injecting the cell with both dyes and before the tissue is put through any histological procedure that might introduce distortion, record an image of the neuron with a fluorescence microscope at high enough magnification to resolve the diameters and relative position of dendritic branches at several positions in the cell. (Note that it is safest to make exposures after a light paraformaldehyde fixation, since illumination of dye-filled cells can result in photoactivated damage and artifactual blebbing.) (3) Record the corresponding images after the complete histological protocol. Shrinkage can then be measured directly and used to obtain a set of scaling factors for correction of model parameters.

One method that has proved successful uses a filling solution with a combination of biocytin and carboxy-fluorescene in the recording electrode. An image of the neuron is obtained immediately following an experiment, using a scanning laser microscope. The confocal microscope is optimal for this purpose since high-resolution, high-magnification images of dendrites relatively deep in unfixed tissue can be obtained, and the net illumination can be kept low enough to avoid photodamage and consequent morphological distortion. Alternatively, tissue can be lightly fixed (i.e., 5 min in 4% paraformaldehyde) and observed with conventional fluorescence microscopy. After recording of several sample images, the tissue is fixed and incubated with HRP-conjugated avidin. The avidin–biotin–HRP complex is subsequently processed through the standard DAB reaction protocols to yield a densely stained cell.

Selection Criteria for Choosing Cells to Reconstruct

To obtain the best results from reconstructions of stained neurons, selection criteria should be used in choosing cells for further analysis. Neurons with filling artifacts such as swollen processes, or faintly stained distal processes, should not be used for reconstructions. Only uniform labeling will facilitate the digitization process and the accuracy of the resulting reconstruction. After selecting a well-labeled neuron, the investigator must consider the issue of cut branches. Although invertebrate neurons

that are labeled in intact ganglia do not exhibit cut processes, the slicing procedures needed to prepare vertebrate tissue invariably result in severed dendrites. If a neuron has been stained in the animal and the tissue sectioned afterward (such as occurs with GFP-expressing neurons or in vivo injections), then it may be possible to locate the remaining portion of a cut dendrite in an adjacent section. However, if a neuron is labeled in a slice (such as with biocytin injections in *in vitro* preparations), then it is quite likely that cut dendrites will be present. To deal with this problem, most investigators develop criteria specifying a maximum number of cut dendrites and their locations. For example, one might specify that a labeled cortical pyramidal neuron will be reconstructed if (1) none of the primary apical dendrites are severed; (2) none of the primary basal dendrites are cut within 100 μm of the soma; and (3) there are ten or fewer cut dendrites on the entire neuron. The criteria are usually devised in accordance with the objectives of the experiment and should take into account previous data on the cell type of interest. An alternative approach to compensate for the cut dendrites is described in chapter 9.

8.2 Methods for Obtaining Anatomical Data at the Ultrastructural Level

For modeling studies where accurate measurements of dendrites, spines, synapses, and subcellular organelles are required, serial section transmission electron microscopy (ssTEM) is the method of choice. The resolution of ssTEM is needed to distinguish and measure dendrites, axons, glial processes, synapses, and subcellular organelles such as synaptic vesicles, polyribosomes, and endosomal compartments. ssTEM is a labor-intensive enterprise; hence, choosing when to use it is important. Confocal and multiphoton microscopy provide lower-resolution images from living cells to assess whether labeled structures have changed location or size; however, the resolution is too low to identify, quantify, and measure the dimensions and composition of structures in the brain neuropil. ssTEM is required for accurate identification and measurement of objects smaller than 250 nm. High-voltage electron tomography coupled with manual segmentation of cellular structures also produces excellent results, although the size of the sample area is much less than with ssTEM, and this approach still requires time-intensive reconstruction and analysis (Soto et al., 1994; Sosinsky et al., 2005).

Another method has recently been added to the electron microscopy toolkit: serial block-face scanning electron microscopy (Briggman and Denk, 2006). This approach has the advantage that fragile serial thin sections need not be handled; instead, the remaining block face is photographed after each section is removed. Theoretically, larger sample areas could be photographed without montaging; however, the resolution achieved at the lower magnification does not allow deciphering of individual

axons, astroglial processes, dendritic spines, synapses, and the composition of organelles in brain neuropil.

Transmission electron microscopy (TEM) usually requires that sections through biological specimens be less than 100 nm thick. These thin sections allow electrons to pass, being stopped only where stains delineate objects of interest. A sample area of 500–1,000 μm^3 spanning 250–500 serial thin sections is practical with ssTEM. Here we describe methods optimized to produce uniform ribbons of serial thin sections. The goal is to produce sections of perfectly uniform thickness along fold-free ribbons for accurate quantitative analyses. In Harris et al. (2006) we describe detailed methods for obtaining large numbers of ultrathin serial sections for 3-D reconstructions of dendrites, spines, and other structures at high resolution. The main steps are briefly described here and a detailed protocol can be found in the original paper.

Tissue Preparation

Rapid fixation of the tissue is essential for the preservation of ultrastructure and can be achieved either through perfusion of the animal or immersion of the tissue in fixative combined with brief microwave irradiation with aldehyde fixation (Jensen and Harris, 1989). This last method produces excellent results and is especially good for fixation of brain slices. Slices of brain tissue (200 μm thick) are immersed in aldehyde fixative (6% glutaraldehyde and 2% paraformaldehyde in 0.1 M cacodylate buffer with 2 mM CaCl_2 and 4 mM MgSO_4) and irradiated briefly (10 s) in a microwave oven, restricting final temperature to less than 40 °C. The fixed slice may be processed immediately or stored overnight in fixative at room temperature. The next day, the slices are first rinsed in cacodylate and then transferred to 0.1 M phosphate buffer and vibra-sliced to 70 μm . Storage of the 70 μm vibra-slices in buffer for more than a day can have detrimental effects on tissue fixation because the fixative is diluted out. The result is a blurring of membranes and darkening of cytoplasm.

Microwave-enhanced staining and dehydration followed by manual resin infiltration appear to be optimal from the perspective of saving time and achieving complete infiltration of the resins. Vibra-slices surrounding the region of interest are postfixed in reduced osmium (1% OsO_4 /2% KFeCN in 0.1 M cacodylate) for 5 min, rinsed in cacodylate buffer, and transferred to 1% OsO_4 in 0.1 M cacodylate buffer in the microwave at 175 W under vacuum with the microwave on for 1 min, off for 1 min, and then on again for a total of 3 min. Sections are then dehydrated at 250 W under vacuum through a series of ethanol dilutions, with all dilutions containing 1% uranyl acetate (UA) to enhance polyribosome staining. Infiltration with epoxy resins is done outside the microwave oven on a rotator in 1:1 and 1:2 ethanol to propylene oxide (PO) for 10 min each; two changes of 100% PO for 15 min each; 1:1 PO:epoxy resin for 1 hr, and 1:2 PO:epoxy resin overnight. The next morning the blocks are transferred through three changes of 100% epoxy resin for 1 hr each and then

properly oriented in coffin molds with fresh 100% epoxy resin and cured for 48 hr in a conventional oven at 60°C. All mixtures with epoxy resins contain the 2,4,6-tridimethylaminemethyl phenol (DMP-30) accelerator. These procedures produce epoxy blocks of uniform infiltration and sufficient hardness to obtain thousands of serial ultrathin sections.

Serial Section Techniques

A section thickness of 45–50 nm is optimal for minimizing overlap between small structures, such as synaptic vesicles (~30–60 nm) or narrow axonal or astroglial processes (~50 nm) that would be obscured by neighboring structures in thicker sections. Care should be taken to start with a block shaved into a trapezoid with the following shape: a height of about 30 µm, a width of about 100–200 µm, and a depth of 20–30 µm for stability and with one side slanted for orientation. Ribbons of serial thin sections are cut and then retrieved on Pioloform-coated slot grids. Lowicryl and some other resins are notoriously difficult to ribbon because the hydrophilic sections fall apart; however, a few quick sprays of salon-quality hair spray and overnight drying produce uniform continuous ribbons on the same trapezoid.

Using a 35-degree diamond knife will minimize the compression of sections. The ribbon will curve if the north and south edges of the trapezoid are not parallel. Uneven thickness of sections is easily corrected by enclosing the ultramicrotome to prevent air drafts and local temperature changes. Stopping and restarting the ribbon can also produce a change in section thickness. A video camera mounted near the microtome can be used to monitor progress. It is essential not to open the enclosure doors until the desired ribbon length has been achieved. These methods produce long, fold-free, and clean ribbons of serial thin sections with a uniform section thickness. Updates on these methods can be found at <http://synapse-web.org>, along with detailed methods of photography, alignment, reconstruction, and measurement in three dimensions.

8.3 Reconstruction Techniques for Cells from Whole Mounts and Confocal Images

To date, the most accurate way of obtaining high-quality 3-D reconstructions is to manually trace the structure of the neuron using software programs developed for that purpose (Capowski, 1985; Jacobs and Nevin, 1991), and the most widely used of these programs is the Neurolucida system (Glaser and Glaser, 1990; Microbrightfield, Colchester, VT; <http://www.mbfbioscience.com/neurolucida>). The branching structure of the neuron is captured as a finite set of points that represent the 3-D coordinates, diameter, and connectivity to other points in the reconstruction. This vector-based format is easy to store, share, and archive, and contains all of the infor-

mation about the geometry of the neuron. Neurons digitized in this format can be used for morphometric analyses as well as modeling studies.

There are two basic techniques used to reconstruct neurons at the light microscopy level: direct tracing from whole-mount preparations in which the entire cell is contained in a thick section or a ganglion, or tracing from a series of confocal images of a fluorescently labeled cell.

Whole-Mount Reconstruction Techniques

Neurons labeled with any of the compounds that result in a dense product, including biocytin, biotinylated dextran amine, cobalt chloride, or the Golgi method, can be digitized directly from whole mounts using a computer–microscope system.

Overall, manual tracing in 3-D is a time-consuming, subjective process. The accuracy and consistency of the reconstruction process depends on the individual collecting the data. Regardless of the functionality of the software or the reconstruction system itself (mechanical stages, microscope, etc.), the quality of the reconstruction depends on how rigorous the operator is when encoding the dimensions of the structure. When neurons are digitized directly from whole mounts, most investigators have found that a $60\times$ or $63\times$ objective provides sufficient resolution for tracing fine neuronal processes and has a working distance that allows successful imaging of processes located toward the middle of the slice (Rihm and Claiborne, 1990).

For complex vertebrate neurons, it is best to digitize the cell body first and then each major dendrite that exits from the soma. Although there are conversion programs (described later) that will convert the digitized neuron into isopotential compartments, care should be taken to ensure that each digitized segment does not change in diameter along its length. Datapoints should be taken at least every $5\text{ }\mu\text{m}$ and at additional points to capture branch points and curves in the branches (Claiborne, 1992). The diameter of the cursor used to trace each branch should be matched to the diameter of the dendrite as closely as possible, but where it is difficult to obtain accurate diameter measurements for the thinner dendrites, measurements taken at the ultrastructural level can be used to correct the diameter measurements. The Neurolucida system allows the operator to label specific parameters, including branch points, natural terminations of neuronal processes, and dendritic spines. In addition, when an entire neuron is not captured in a single thick section, the severed dendrites at the edge of one section should be marked as such in the data file. If additional sections are available, they can be lined up with the severed ends of the same processes. These labeled dendrites can then be digitized and added to the tree.

Similar techniques can be used to digitize invertebrate neurons, with the caveat that some of these cells have much larger and more complex dendritic trees than do hippocampal pyramidal neurons or cerebellar Purkinje cells, with reconstructions containing up to 10,000 segments compared with 1,500–3,000 for vertebrate neurons

(Jacobs and Theunissen, 1996, 2000). In previous studies, using a reconstruction system designed in the Jacobs lab (Jacobs and Nevin, 1991), two different techniques for reconstructing large interneurons and sensory neurons were developed. For interneurons, the cell body was digitized first, followed by the major branch points along the primary neurite to the axon. When a termination point was entered, the software would automatically move to the last open termination point and the subsequent branches could be digitized. Thus each major branch could be digitized in reverse order, from the axon back to the cell body. This approach is very useful in that as long as the operator does not miss any branch points, the software program will keep track of all open termination points, thus ensuring that no major portions of the tree are missed. These reconstructions took up to 50 hr to complete in some cases.

A simpler method can be used to digitize parts of the neuronal structure. For example, in studies where the number, distribution, and location of (synaptic) varicosities were measured and modeled (Jacobs and Nevin, 1991; Jacobs and Theunissen, 1996, 2000), the primary branching structures of the sensory afferent axonal terminals were digitized and the varicosities entered as individually labeled points. These reconstructions captured the main branching structure of the axon terminal and the distribution of varicosities. These partial reconstructions can be completed in approximately 3 hr, a huge savings of time compared with doing a complete reconstruction. Although these reconstructions cannot be used to model the electronic properties of these cells, the location, size, type, and distribution of the varicosities can be used to model synaptic contacts onto postsynaptic neurons (Jacobs and Theunissen, 2000; Cummins et al., 2003).

Reconstructions of Neurons from Confocal Images

Recent developments in both commercial (Neurolucida) and open-source software (Neuron_Morpho, described later) now provide methods for reconstructing fluorescently stained neurons from stacks (Z series) of confocal images (K. M. Brown et al., 2005). A series of images can be obtained from the preparation, stored, and reconstructed offline. Previous attempts to reconstruct fluorescently labeled cells directly from thick tissue sections as described here have been less successful, owing to fading of the label under long illumination times. Digitizing directly from a stack of confocal images avoids bleaching the dye and allows the reconstruction to be performed on any computer. This can be done either with the Neurolucida software using the confocal module or with the Neuron_Morpho software application described later.

For neurons labeled with fluorescent dyes such as DiI or GFP, a series of overlapping stacks of images (with 0.5- μm steps between scans) are acquired with a confocal laser scanning microscope using a 40 \times objective. Each stack covers a portion of the tree, and although the exact amount of overlap needed will depend on the qualities of

the objective, an overlap of about 50 μm is usually sufficient. The number of image stacks will depend on the size of the dendritic tree; for example, hippocampal pyramidal neurons in mice usually require three to five stacks. The stacks are first montaged together and then the entire dendritic tree is digitized. Because of the resolution of the confocal images and the computer monitor, it can be difficult to obtain accurate diameter measurements for the thinner dendrites. A sampling of dendritic diameters can be measured with the confocal microscope, followed by correction of the data files (O'Boyle et al., 1993). When labeled neurons have dendrites extending into adjacent sections, the severed dendrites at the edge of one section can be marked in the data file and then lined up with the cut ends of the same processes in the next section.

Giorgio Ascoli and his colleagues have developed an open-source tool *Neuron_Morpho* (K. M. Brown et al. 2005; <http://www.personal.soton.ac.uk/dales/morpho/>) that allows the accurate reconstruction of neurons from confocal image stacks. This tool works as a plug-in for the popular ImageJ (<http://rsb.info.nih.gov/ij/>) open-source tool developed and distributed by the National Institutes of Health (Bailer, 2003). Both of these tools are free and are compatible with all major computer platforms (Windows, Mac, and Linux). *Neuron_Morpho* takes advantage of the familiar functionality of ImageJ image handling, yet provides mouse-actuated tracing of dendritic processes. Another useful feature is the “return” function that is activated when an end point to a dendritic branch is entered. The program automatically scans back to the most recent branch point.

First, a set of tiled optical sections that encompass the entire extent of the dendritic tree is collected and then merged into a single dataset using VIAS (Volume Integration and Alignment System; <http://www.mssm.edu/cnic/tools.html>), a freeware tool that automates the alignment of confocal images (Rodriquez et al., 2003). The merged image stacks are then loaded into ImageJ for manual reconstruction and the files converted into the standard format, Stockley-Wheal-Cannon (SWC) (described later) for import into either NEURON or GENESIS modeling programs (see the software appendix). The reconstruction can be calibrated by imaging a calibration grid and measuring it using the length tool in ImageJ.

A comparison of these tools with Neurolucida (using the confocal module) demonstrated no loss in accuracy and only a small difference in ease of use (K. M. Brown et al., 2005). Of interest, however, was the observation that neurons reconstructed using *Neuron_Morpho* have many more segments than the same neurons reconstructed using Neurolucida, although both programs produce reconstructions of similar accuracy. This may be due to the differences in the digitization process: in *Neuron_Morpho*, segments are delineated with vertical lines drawn on either side, whereas in Neurolucida, a circular cursor is used to mark the boundaries.

General Sources of Measurement Error in Light-Microscopic Measurements of Cell Dimensions

Regardless of the method used to make the reconstruction, there are several significant intrinsic practical limitations to light microscopic measurements of neuronal structures. The finest dendritic processes, with diameters less than about 0.5 μm , cannot be detected adequately for reconstruction and will be missed. Fine processes between about 0.5 and 2 μm can be resolved, but the diameters cannot be measured accurately with conventional (wide field) light microscopy. In practice, owing to the characteristic point spread function of a particular optical system, such fine branches will appear larger in diameter than their true size. Very fine in-foldings and undulations in dendritic processes cannot be resolved, leading to a systematic underestimate of dendritic surface area.

Although these problems would seem to be very serious, their effects upon simulated cell responses can be estimated and corrected to a large extent. The key, as mentioned earlier, is to make high-resolution measurements of sample areas within the tissue using a confocal microscope, or preferably, using correlated light and electron microscopy. For example, the high-resolution images can be used to measure the density and morphological parameters of spines, varicosities, and/or in-foldings in dendritic segments of a range of different diameters and locations. These true values can be compared with the low-resolution values extracted through a light microscopic analysis to obtain systematic correction factors, which can be applied to the anatomical data before computer model parameters are set. For example, cortical neurons can be labeled and reconstructed for dendritic measurements and then samples of dendritic segments can be re-embedded for electron microscopy (Megías et al., 2001). Accurate spine density measurements taken from serial sections then can be incorporated into the resulting data files and computational models.

Another approach is to estimate and quantify the impact of these errors by running simulations of the model when several orders of very fine dendritic branches are added onto or deleted from the model's structure. The extent to which such operations change a simulated response depends on the type of question asked by the simulation. If one is calculating the complex input impedance at a point on a fine branch of a dendrite, and one adds or subtracts extra branches near that point, then the calculations could change significantly. If, however, one is calculating the complex transfer impedance between the bases of two different second-order dendritic branches, then the addition or deletion of several fifth- or sixth-order branches has an insignificant effect.

The same general approach can be used for errors in surface area measurements that are due to subresolution of fine structure: errors are estimated, correction factors are used, and the model's tolerance is estimated by bracketing simulations. The

errors of the second type given here (i.e., the misrepresentation of process diameters in the 1–2-μm diameter range) are more problematic. Using confocal microscopy and reconstructing the cell from image stacks as described earlier could minimize these errors. Dendritic processes could be reconstructed down to the 2-μm level, which is well above the level of ambiguity for a typical confocal scope. All processes below 2 μm could be dealt with statistically from electron microscope observations (see Megías et al., 2001).

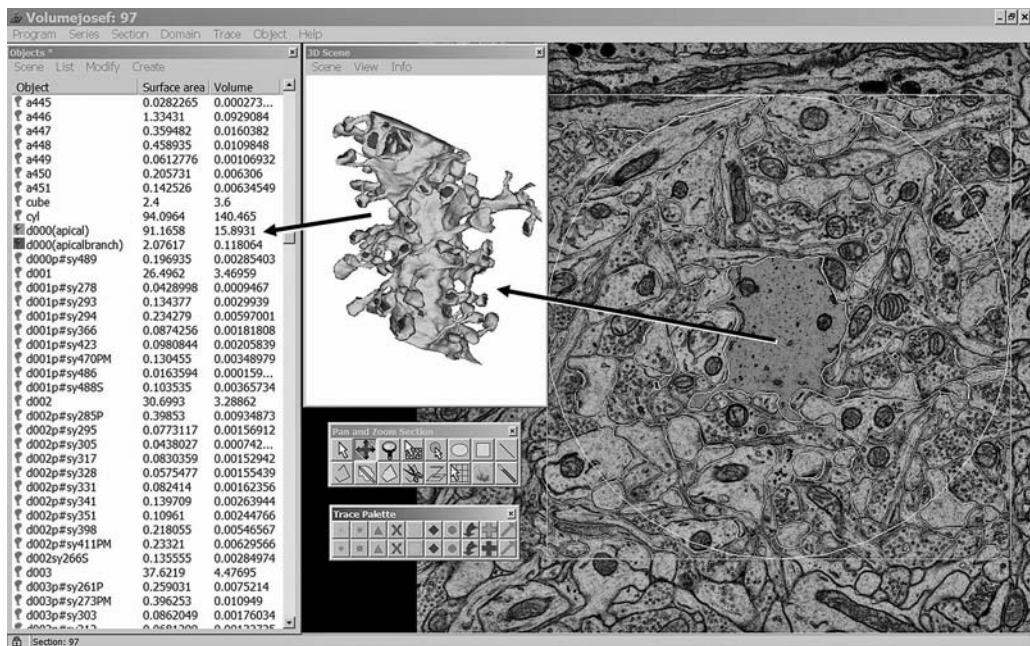
Ultrastructure Reconstruction Techniques

Serial section electron microscopy provides the best resolution for neuronal structure at both the cellular and subcellular levels. A suite of open-source software tools, *Reconstruct* (Fiala, 2005; <http://synapse-web.org/>), has been developed by Fiala and Harris (2002) to aid the manual segmentation of neuronal structures of interest from serial electron microscope sections. Many microscopy studies require reconstruction from serial sections, a method of analysis that is sometimes difficult and time-consuming. When each section is cut, mounted, and imaged separately, section images must be montaged and realigned to accurately analyze and visualize the three-dimensional structure. Reconstruct is a free editor designed to facilitate montaging, alignment, analysis, and visualization of serial sections (figure 8.2 shows the interface for Reconstruct). The methods used by Reconstruct for organizing, transforming, and displaying data enable the analysis of series with large numbers of sections and images over a large range of magnifications by making efficient use of computer memory.

The alignment of images from sections with some types of nonlinear deformations, including cracks and folds, can be corrected, although when sections are handled carefully as described in Harris et al. (2006), folds and cracks can be avoided. A large number of different structures can be easily traced and placed together in a single 3-D scene that can be animated or saved. A number of other software packages are available to support structure-oriented reconstruction from serial sections, but do not provide the full functionality of Reconstruct. Some of these commercial programs (e.g., 3-D Doctor, Amira, Neurolucida) provide algorithms for automatic alignment of an entire stack of sections. Automatic alignment plug-ins are also available for the public domain image-processing package ImageJ. However, the quality of these “autoalign” functions needs to be carefully monitored before using them for reconstruction.

An excellent detailed tutorial and documentation of the Reconstruct program has been written by John Fiala. There are six basic steps to this process, described briefly here:

1. First the operator creates a series by designating a directory where the files for the reconstruction will be stored.

**Figure 8.2**

User interface for the Reconstruct software program. The left part of the image shows quantitative data for each of the traced contours in the reconstruction. The center field shows a section with traced profiles and the center filled profile is a section of the reconstructed dendrite shown to the left.

2. Then a set of sections are imported as image files in formats that include Windows bitmaps, GIF, TIFF, and/or JPEG files. Each image file should contain only a single image (not an image stack) and represent one section in a series of eight or twenty-four bits per pixel.
3. The next step is to calibrate the pixel size to specify the size of the image picture elements (pixels) in terms of the units of measure of the series. In most cases the pixel size should be adjusted by using the image of a calibration grid or scale that was photographed with the series. Section thickness should also be calibrated using the cylindrical diameters method (Fiala and Harris, 2001).
4. Then the sections in the series must be aligned. Tissue that has been physically sectioned and then imaged must be realigned for accurate 3-D measurements and visualizations. Section images can be manually moved into relative alignment using keyboard commands; however, the software supports a semiautomatic method where the operator specifies correspondence points in two adjacent sections and the program aligns them to each other.

5. The next step is to trace the profiles of the structures of interest within the sections. The operator may chose to reconstruct a single dendrite or organelle, or specify multiple objects and create an ensemble reconstruction showing the relationships between neuronal structures within the scene. Contours are drawn with a pencil tool and filled with a color when the contour is closed. The process is repeated throughout the series of images.

6. At any time during the reconstruction, the operator can view the 3-D object that is being traced. Objects can be viewed in any combination and the scene window supports rotation, pan, and zoom. The objects may be viewed in wire frame or surfaced using a Boissonnat surfacing algorithm. These scenes can be saved for publication as bitmaps or as virtual reality modeling language (VRML) objects for additional smoothing, lighting, and rendering in other 3-D programs (e.g., DReAMM, 3-D Studio Max, Amira etc).

The software can be obtained from the <http://synapse-web.org> site, which also provides tutorials, extensive documentation, and numerous examples of reconstructed dendrites, spines, and subcelluar structures.

3-D Reconstruction of Cellular Ultrastructure Using Electron Tomography

High-voltage electron tomography is an alternative method that can be used to obtain reconstructions of subcellular anatomy. This method produces a series of images that can be manually segmented using a combination of commercial and open-source software tools. Mark Ellisman's laboratory at the National Center for Microscopy and Imaging Research (NCMIR) has pioneered the development of these tools. The methods for preparing samples, tomographic imaging, and alignment of images are described in depth in a recent modeling study on the node of Ranvier (Sosinsky et al., 2005) that uses the anatomical data obtained through a tomographic reconstruction as the basis for a simulation study using the MCell modeling environment (see the software appendix). Examples of tomographic data, including all of the data used in the node of Ranvier study, can be viewed and or obtained from the Cell Centered Database (see section 8.5).

Investigators at NCMIR have developed a set of techniques and an extensive set of tools for 3-D reconstruction of cellular components from tomograms. These software tools are available for free (<http://ncmir.ucsd.edu/downloads/software/index.shtml>). The following is a brief description of these techniques and the tools used.

Features of interest (membrane regions, organelles, and cytoskeletal elements) can be traced from the tomographic data using *Xvoxtrace* (Soto et al., 1994) and rendered using the SYNU 3-D graphics package (Hessler et al., 1992). *Xvoxtrace* enables the user to use manual tracing techniques for volume segmentation of

tomographic data. Using Xvoxtrace, one can outline features on individual planes of the volume while being guided by simultaneous views of the tracing displayed on a volume rendering or tilt series. Once the contours are traced, they can be viewed using Dend or used to generate surfaces for viewing in SYNU. SYNU works with polygon meshes, stacks of contour lines, or 3-D volumes from confocal Z-series, tomography, or simulations. Graphical representations that combine volume and surface renderings can be computed and visualized using the commercial program Amira (TGS Inc., San Diego, CA; <http://www.amiravis.com/>). Anatomical data obtained through reconstruction of tomographic images can be imported into DReAMM and used with MCell for modeling studies (see the software appendix).

8.4 Using Accurate Neuronal Morphology in Neuron Models

Chapters 10–11 describe how the structural parameters captured with 3-D reconstruction techniques are combined with biophysical properties to create a model. However, once the reconstruction is complete, the file containing the anatomical parameters must be transformed into formats compatible with modeling platforms such as GENESIS or NEURON (see the software appendix). There are several software tools that provide this functionality: Neurolucida, CVapp (R. C. Cannon et al., 1998), L-measure (Ascoli et al., 2001b), and MIEN.

The Neurolucida system contains an analysis module called *NeuroExplorer*, which has an extensive set of functions for visualizing, editing, and performing morphometric analyses on the reconstructed cells. The program supports 3-D visualization and morphometric analysis and can be used to dynamically rotate, scale, and set the display characteristics of any Neurolucida data file. NeuroExplorer can be used to edit the reconstruction and correct errors in digitizing prior to converting the file into a format for modeling. NeuroExplorer exports files only in the Neurolucida format.

The computational neuroscience community has developed several open-source tools that have much of the same functionality as the Neurolucida system, but are specifically designed to edit and convert the morphological information for use in modeling programs. Two of these programs, CVapp and L-Measure, have been in wide use for some time. MIEN is a new addition to the toolbox. These software suites and their attributes are described in the following paragraphs.

CVapp (R. C. Cannon et al., 1998; <http://www.compneuro.org/CDROM/docs/cvapp.html>) written in Java, is a visualization, format conversion, and manual editor for digital reconstructions that can convert files from either the Neurolucida or SWC format into either the NEURON or GENESIS format for modeling. The SWC format, developed by Robert Cannon, is a parsimonious format widely used for neuronal

morphologies. Each point is represented by seven numbers corresponding to its unique identity; neuritic type (soma, dendrite, axon); X, Y, and Z position; radius; and the identity of its parent in the path from the soma. CVapp supports a number of editing and viewing functions: error checking, shrinkage correction, and adding or deleting branches. It supports multiple viewing styles and can be run through the web as an applet or installed on the user's machine.

L-Measure, developed by Giorgio Ascoli and his colleagues (Ascoli et al., 2001; <http://krasnow.gmu.edu/L-Neuron/L-Neuron/home.htm>) allows researchers to extract quantitative morphological measurements from neuronal reconstructions. This tool has some of the same file conversion and editing functionality of CVapp and will convert files from Neurolucida and Neuron_Morpho into the SWC standard format. Given the idiosyncratic, error-prone nature of the digitizing process, L-Measure has been used as a tool to standardize 3-D reconstructions from many investigators. This tool recognizes and fixes irregularities in the reconstruction and flags others for manual inspection and repair. L-Measure also contains a powerful, comprehensive suite of morphometric analysis tools that can be applied to any reconstruction for the quantitative analysis of neuronal structure. Characteristics such as numbers of compartments or branches, total dendritic length, surface area, volume, and a wealth of branching characteristics can be calculated with this tool. Both CVapp and L-Measure have become essential tools for converting "imperfect" 3-D reconstructions into models as well as the quantitative analysis of neuronal structure.

MIEN (Model Interaction Environment for Neuroscience), developed by Graham Cummins (<http://mien.sourceforge.net>), provides a framework for storing, integrating, and interacting with neuroscience data, including data on anatomy and physiology, abstract mathematical models, and detailed compartmental models. MIEN is not a compartmental model simulator, but it provides an interface to the NEURON simulator for evaluation of compartmental models. MIEN supports editing and display of 3-D reconstructions in the Cell Viewer (figure 8.3). This module provides fully 3-D interactive representations of cells, fiducial marks, synapses, density clouds, and annotations. The viewer uses OpenGL functions to produce quality 3-D images and display them in full 3-D and can simulate the optical sections taken by a microscope with a particular viewing angle and depth of field. It allows user interaction and provides functions for selecting points, sections, paths, and other regions within an anatomical model. The viewer also allows the creation, deletion, and modification of structures and provides tools for spatial alignment (coregistration) of different anatomical datasets using fiducial landmarks or direct measurements of relative size and location. It can also load physiological data and display properties like channel density and membrane voltage by coloring the displayed cells.

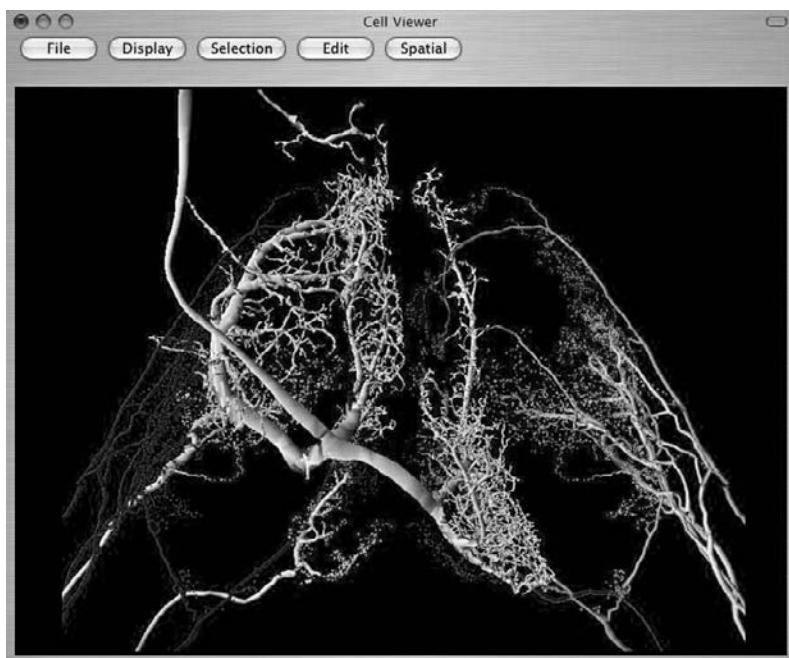


Figure 8.3

Cell Viewer interface from the MIEN suite of software tools. The display shows a cricket interneuron and several mechanosensory afferents that have been scaled and aligned to the same coordinate system. The cells were reconstructed in 3-D and visualized using tools within Cell Viewer.

8.5 Databases and Archives of Neuronal Morphology

Over the past five years, a number of databases and archives of high-quality structural data for the neuroscience community have become available (Ascoli, 2006). Many of these efforts have blossomed through the efforts of investigators supported by the Human Brain Project (Koslow and Hirsch, 2004; Gardner and Shepherd, 2004), and the outcome of this work has produced two comprehensive clearing houses for this type of data as well as many other kinds of neuroscience data: The Neuroscience Database Gateway (NDG) (<http://ndg.sfn.org>, 2004) and more recently, the Neuroscience Information Framework (NIF) (<http://www.neuinfo.org/>). The goal of these efforts is to support and enhance data sharing within the neuroscience community. Although there are many existing collections of anatomical data residing in the laboratories of individual investigators, several community databases, which host anatomical data from many different labs, have now been developed. Three of these efforts have emerged as leaders within the neuroinformatics community as providers of data, tools, and ontologies. These three archives are

described in detail here; additional collections can be found at the NDG and in the future at the NIF.

Cell Centered Database

The Cell Centered Database (CCDB) (<http://ccdb.ucsd.edu/>) was the first web-accessible database for cellular imaging data (Martone et al., 2002, 2003). This database houses structural and protein distribution information derived from confocal, multiphoton, and electron microscopy, including correlated microscopy. The CCDB is affiliated with the National Center for Microscopy and Imaging Research (<http://ncmir.ucsd.edu>), a national research resource specializing in high-voltage electron microscopy, electron tomography, and the development of state-of-the-art 3-D imaging and analysis technologies for structure and function analysis. The CCDB hosts a wide variety of structural information, including images of many cell types and 3-D reconstructions of individual cells, which are stored as series of optical slices, both with and without deconvolution, and also as branching tree structures traced using Neurolucida.

Datasets obtained with electron tomography are stored along with all of the raw images and the processing details required to reconstruct the volume from the raw data. Each object segmented from the 3-D volume is stored as a separate object indexed to the parent reconstruction. Four types of segmented objects are modeled in the CCDB: (1) surface objects, polygonal surface meshes representing 3-D objects in the reconstruction extracted using either isosurfacing methods or manual contouring; (2) contour objects, a series of contours that have not been fitted with a surface; (3) volume objects, subvolumes containing an object of interest; and (4) tree objects, skeletons of branching objects like dendrites and axons derived from Neurolucida. Each object is stored with such measurements as surface area, volume, length, number, and labeling intensity. The CCDB is an excellent resource for many different types of structural data and advanced microscopy tools and techniques, and serves as a testbed for the development of ontologies for neuroscience and data mining techniques.

NeuroMorpho.Org

This effort, led by Giorgio Ascoli and his colleagues (<http://neuromorpho.org>), under the auspices of the Neuroscience Information Framework, is a searchable archive of over a thousand digital reconstructions contributed by investigators in other laboratories. More than twenty-two different laboratories have contributed cells from four or more species of animals, and links are provided to the contributors' websites. At least eight identified cell types are represented, with the majority coming from the hippocampus and neocortex. Each neuron in NeuroMorpho.Org is represented by a unique identifier, general information (metadata), the original and standardized

ASCII files of the digital morphological reconstruction, and a set of morphometric features. Each of these reconstructions has been evaluated and edited using the L-Neuron software. Errors and inconsistencies have been corrected, and all reconstructions are stored in the same format: the standard SWC mentioned earlier. This important and growing database serves as the largest curated archive of 3-D reconstructions and represents a huge resource for the computational neuroscience community.

Synapse Web

Synapse Web (<http://synapse-web.org/>) is a resource developed for understanding and studying the 3-D ultrastructure of the brain. It combines tutorials on the sub-cellular structure of neurons and comprehensive descriptions of procedures and tools for tissue preparation and 3-D reconstruction (described earlier), an atlas of neurocytology (developed by Josef Spacek), several volumes of aligned serial section images of neuropil in the hippocampus, and numerous fully reconstructed dendrites, some with an electron microscope series of images. This resource provides a set of very high quality educational resources, superb tools for reconstruction, and access to high-quality ultrastructural data.

9

An Approach to Capturing Neuron Morphological Diversity

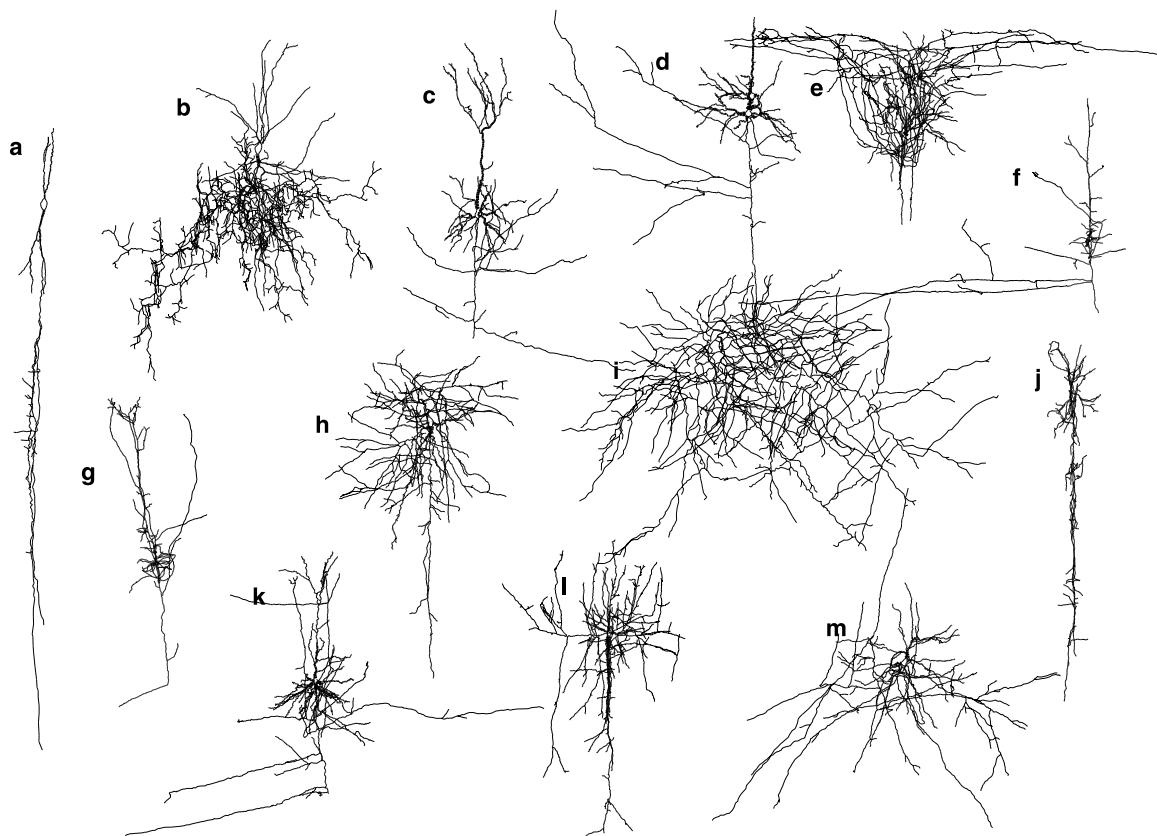
Haroon Anwar, Imad Riachi, Sean Hill, Felix Schürmann, and Henry Markram

9.1 The Diversity of Cortical Neuron Morphologies

Neocortical neurons display a wide range of dendritic as well as axonal morphologies, ranging from compact arborization to highly elaborate branching patterns (Peters and Jones, 1984; figure 9.1). Dendrites play an important role in the spatial and temporal integration of postsynaptic potentials, both metrically (Rall et al., 1992; Rapp et al., 1994) and topologically (Van Pelt and Schierwagen, 1994, 1995; Van Ooyen et al., 2002), and are consequently important determinants of which inputs are collected from local and remote brain regions to determine the characteristic neuronal firing patterns (Larkman and Mason, 1990; Mason and Larkman, 1990; Mainen and Sejnowski, 1996; Sheasby and Fohlmeister, 1999). Axons are crucial for the network connectivity because they show remarkably specific interlayer as well as intercolumnar arborizations that enable focused transmission of information. Together, dendritic and axonal morphologies underlie many key aspects of cortical function, including the diversity of neuron electrical properties, the extent of neural connectivity, and ultimately the integration and propagation of information within and between neocortical columns.

Modeling the Diversity of Cortical Neuron Morphologies

In order to achieve the goal of capturing neuron morphological diversity, whole-cell morphologies must be obtained for each of the many known morphological classes of neurons. Cell reconstructions are subject to artifacts that affect morphological shape and size as a result of slicing damage or shrinkage from the histological process (chapter 8, section 1). Single-cell models are sensitive to the electrical and integrative changes caused by such damage, while the specific morphological shapes of dendrites and axons strongly influence the connectivity in a network simulation. To address this issue, we develop and test an algorithm for repairing computer-based neuronal reconstructions.

**Figure 9.1**

Morphological diversity of cortical neurons: (a) Bipolar cell. (b) Chandelier cell. (c) Layer 4 pyramidal cell. (d) Star pyramid cell. (e) Martinotti cell. (f) Corticocortical cell. (g) Corticothalamic cell. (h) Nest basket cell. (i) Small basket cell. (j) Double bouquet cell. (k) Layer 2 pyramidal cell. (l) Large basket cell. (m) Spiny stellate cell. (Neurons not drawn to scale.)

Limitations of Current Methods for Reconstructing Whole-Cell Morphologies

Neuron morphologies can be reconstructed from stained neurons obtained in two kinds of experimental techniques: *in vivo* and *in vitro* (see chapter 8, section 3). Although a morphology reconstructed from neurons obtained *in vivo* is more accurate than one obtained *in vitro*, the number of neurons that can be harvested *in vivo* and the variety of neuron classes recorded is much more constrained than *in vitro* experiments. On the other hand, data obtained *in vitro* are morphologically incomplete, owing to the slicing of the brain. This problem is more severe for patch-clamp recordings than sharp-electrode recordings since the neurons recorded are closer to

the surface where their upper arborizations are cut. Other limitations in reconstructing the full morphology of neurons are artifacts in the morphological data, due to dye leakage, incomplete filling, tissue shrinkage, and distortion (chapter 8, section 1; Horcholle-Bossavit et al., 2000). Incomplete dye loading can introduce serious artifacts in the morphological representations and morphometric analysis, but shrinkage and distortions can at least be partially corrected.

Another source of potential artifacts, in both techniques, is human inconsistency. This inconsistency is due to the arbitrary sampling of data points; the dependence of the diameter on the reconstruction system and the assignment of continuation, bifurcation, and termination based on subjective decisions (Jaeger, 2001). Automated reconstruction methods are rapidly improving and have the promise to significantly reduce this source of error in neuronal reconstructions.

Computer Generation Techniques for Morphologies

Numerous computer-based techniques have been applied to generate neural cellular morphologies. Generally, these techniques fall into the categories of diameter-dependent algorithms and those based on branch order or branch distance from the soma.

Diameter-dependent algorithms are based on the empirical observation that thicker branches tend to end in bifurcations, whereas thinner branches tend to end in terminations. Probability functions are used to assign the diameters of the bifurcation and the branch lengths (Hillman, 1979; Burke et al., 1992; Ascoli et al., 2001b; Burke and Marks, 2002; Donohue et al., 2002). Several diameter-dependent algorithms are implemented in L-Neuron (Ascoli and Krichmar, 2000; <http://krasnow.gmu.edu/L-Neuron/L-Neuron/home.htm>).

In branch order-based algorithms, the probabilities of bifurcation and termination decrease and increase with the order, respectively (Kliemann, 1987; Devaud et al., 2000; Van Ooyen and Van Pelt, 2002). These algorithms may be considered as variants of the distance-dependent algorithms (Nowakowski et al., 1992; Senft, 1997; Ascoli et al., 2001a), where the probability of bifurcation and termination depends on the path distance from the soma center.

Branch order-based algorithms generate dendritic topology and branch lengths without taking into account segment diameters. However, segment diameters determine important passive properties for neuron modeling (chapter 10, section 1). Furthermore, diameter-based models (such as L-Neuron) are highly sensitive to the initial diameter at the beginning of the “growth” (i.e., close to the soma, where measurement error is usually small), although these models are fairly robust with respect to error in diameters near the terminations (Ascoli, 2002).

An issue with most existing algorithms is that *a priori* information about the true parameter distributions is required but is not available. To address this issue, a

nonparametric algorithm known as KDE (kernel density estimation)-neuron has been proposed (Torben-Nielsen et al., 2008), which faithfully estimates parameter distributions directly from the anatomical data.

Recently, a dendritic growth model has been built on a diffusion-limited aggregation process where the growth is independent of intrinsic geometric parameters (Luczak, 2006). The growth in this model is entirely based on the environmental factors and simple rules of diffusive growth, which also provides an account of the morphological diversity of neurons that is due to the competitive growth among neurons. Despite promising results, the morphologies generated are not yet accurate enough to faithfully represent anatomical data. To address all these limitations, a unified and more robust model incorporating diameter, order, and distance must be developed for dendritic modeling.

9.2 Repair of Neuronal Dendrites

We describe a new probability density cloud approach for characterizing and repairing neuron dendritic and axonal morphologies. This approach is based on a probabilistic analysis of branch characteristics at various distances from the neuron soma. The resulting probabilities are used to reconstruct the missing portion of the sliced neuron's morphology. This approach is then used to repair a layer II/III pyramidal cell and validated using standard morphometric analyses.

Assumptions

The goal of the repair process is to recover the missing parts of the neuron morphology that are lost during the slicing procedure. Though it is not possible to recover exactly the missing morphology, our aim is to recover the missing structure in a statistical manner. Considering the variability of dendritic morphology within a class, the “dendritic repair” process uses characteristics from the intact portion of the neuron to be repaired, rather than the averaged characteristics across many neurons from the same class. The key assumption is that dendrites have an underlying statistical symmetry in their branching features. In contrast, the “axonal repair” process assumes that local axonal arborization (within the bounds of a single cortical column) is not symmetric. Asymmetric axonal branching provides sparse data over the potential volume of a neuron. Therefore the axonal repair process requires the collection of morphological features over many neurons to derive sufficient statistics to reconstruct an axon.

For the purposes of this work, a neuron is represented as a set of connected points where each point is described by a 3-D (x,y,z) coordinate system and diameter at that point. A *branch* is the set of points between two bifurcations, where any two consec-

utive points can be considered a *segment*. The branch originating out of the soma body has a branch order zero, which increases as we traverse each tree to its next level.

A dendritic model evolved from existing models is used in the repair process (Hillman, 1979; Kliemann, 1987; Burke et al., 1992; Nowakowski et al., 1992; Senft, 1997; Ascoli, 1999, 2002; Ascoli and Krichmar, 2000; Devaud et al., 2000; Ascoli et al., 2001a, 2001b; Burke and Marks, 2002; Donohue et al., 2002; Van Ooyen and Van Pelt, 2002) for the anatomical modeling of dendrites. This model defines the behavior of the branches given the branch order and distance from the soma. The behavior of the branches can be classified as continuation of a branch, bifurcation of a branch, or termination of a branch. These probabilities therefore define the characteristics of a neuron and form a dendritic model. Another important characteristic of a neuron is the diameter of the branches. Although the diameter is not explicitly included in the model owing to the variability (measurement error), it remains an important parameter that is implicitly taken into account during the repair process, as discussed later.

The formalization of this model is termed the probability density cloud and represents the spatial distribution of dendritic branch properties as probabilities within spherical shells centered at the cell body and with progressively increasing radii. Starting from this definition, we divide the 3-D space into spherical shells to analyze the properties of the dendrites within each region in space. These shells are related to the spatial divisions used by Sholl (1953) in analyzing the density of dendritic branches as a function of distance.

The main assumption underlying the repair algorithm is the statistical symmetry of a neuron, in contrast to the geometrical symmetry. In the case of geometrical symmetry, there is an exact correspondence of the morphology on opposite sides of a plane or about an axis. However, in the case of statistical symmetry, the probabilistic parameters controlling the growth of the dendrites are symmetrical about a point or axis, not the morphology.

Each spherical shell defines a portion of an overall probability cloud in terms of the probability of a termination, bifurcation, and continuation event at each order. Since these probabilities are order dependent, the probability functions are given as Bayesian probabilities (i.e., probability of an event E_i occurring knowing that the branch is of order O_i). $P(E|O, S)$ is the probability of event E occurring knowing that the branch is of order O and is in shell S of the probability cloud.

To calculate these probabilities for all branches of order O at shell S , we calculate the fraction of branches for each event E :

$$P(E|O, S) = \frac{N_{E,O,S}}{N_{O,S}}, \quad (9.1)$$

where $N_{E,O,S}$ is the number of branches of order O undergoing event E at shell S and $N_{O,S}$ is the number of branches of order O at shell S .

The intershell spacing is constant and has been chosen so that the spacing can accommodate the mean distance required for a bifurcation. The precise choice of the intershell distance did not significantly affect the repair, with variations in morphometric parameters of less than 2%.

The decision for growth is made explicitly using probabilities specific to the order and shell, and implicitly on the distance from the soma (which serves as the symmetry point). Thus, patterns of branching at specific distances as well as the maximum reach of growth are addressed based on the statistics of the intact neuron.

Unraveling Dendrites

Tissue shrinkage can lead to a decrease in the overall reach of an individual neuron (chapter 8, section 1). At the same time, the neurites increase in tortuosity, although their overall length is assumed to stay the same (Jaeger, 2001). To correct for this increase in tortuosity, we have developed an unraveling procedure that smoothes the neurites while retaining their overall length.

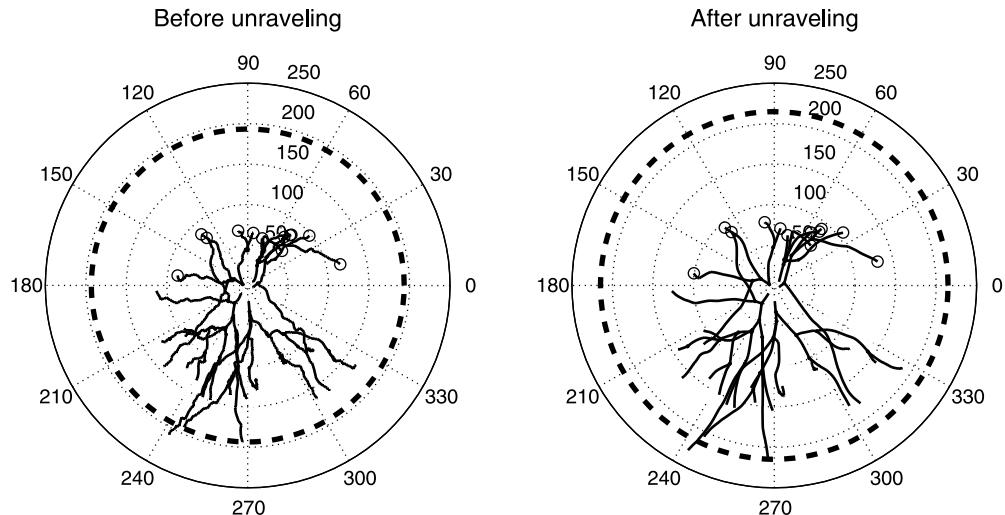
The unraveling is performed using a moving window algorithm for each branch: assume that the window is of size N and that it is running on a branch of M points. The window is centered on the first point of the branch to be corrected (call this point's index p), taking the $N/2$ previous points and $N/2$ next points. Hence all the points between $p - N/2$ and $p + N/2$ are fitted by a 3-D direction vector \vec{D}_p using principal component analysis (PCA). At the boundaries (if $p < N/2$ or $p + N/2 > M$) the window is truncated.

The moving window runs first on the tortuous neuron, computing all the direction vectors for each segment in the branch, and then updates the directions of the segments. Starting from the first one and moving to the end, the segments are aligned with the main direction computed for that point and the coordinates of the point $p + 1$ then become:

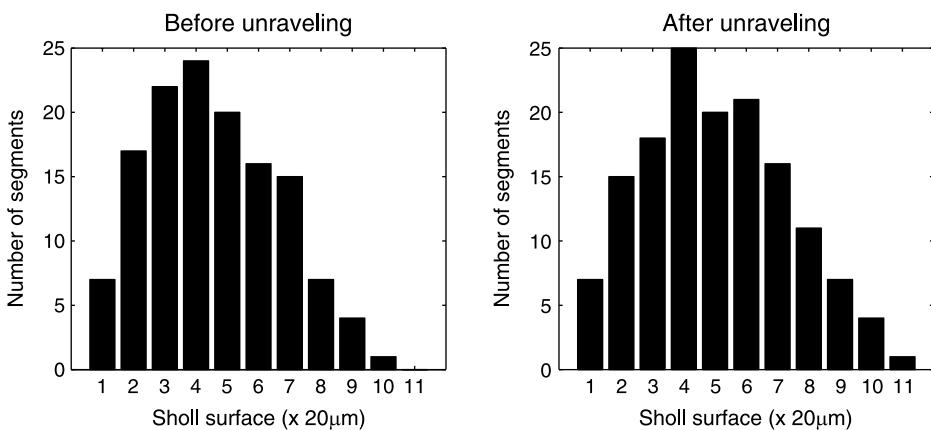
$$\begin{pmatrix} \hat{x}_{p+1} \\ \hat{y}_{p+1} \\ \hat{z}_{p+1} \end{pmatrix} = \begin{pmatrix} \hat{x}_p \\ \hat{y}_p \\ \hat{z}_p \end{pmatrix} + \|\vec{V}_p\| \frac{\vec{D}_p}{\|\vec{D}_p\|}, \quad \vec{V}_p = \begin{pmatrix} x_{p+1} - x_p \\ y_{p+1} - y_p \\ z_{p+1} - z_p \end{pmatrix} \quad (9.2)$$

The unraveling preserves the overall length of the neuron. Its window size can be adjusted to achieve a particular increase in the reach of the neuron as well as a decrease in the tortuosity of the segments (more smoothing).

The unraveling process extends the reach of the neuron away from the soma center, as shown in figure 9.2. This increase in extent is also revealed in the Sholl distribution of basal dendrites before and after the unraveling (figure 9.3). The window

**Figure 9.2**

Comparison of spatial mapping of dendritic points on XZ plane before and after unravelling. This layer 2 pyramidal cell is unraveled using a window size of ten morphological points. The thick black dotted line depicts the computed morphological bounds used during repair. Open circles denote the identified cut points of the dendrite.

**Figure 9.3**

Comparison of spatial characteristics (Sholl distance) of dendrites before and after unravelling. This layer 2 pyramidal cell is unraveled using a window size of ten morphological points.

size influences the increase in extent and should be fitted to the amount of shrinkage observed that is due to the histological preparation.

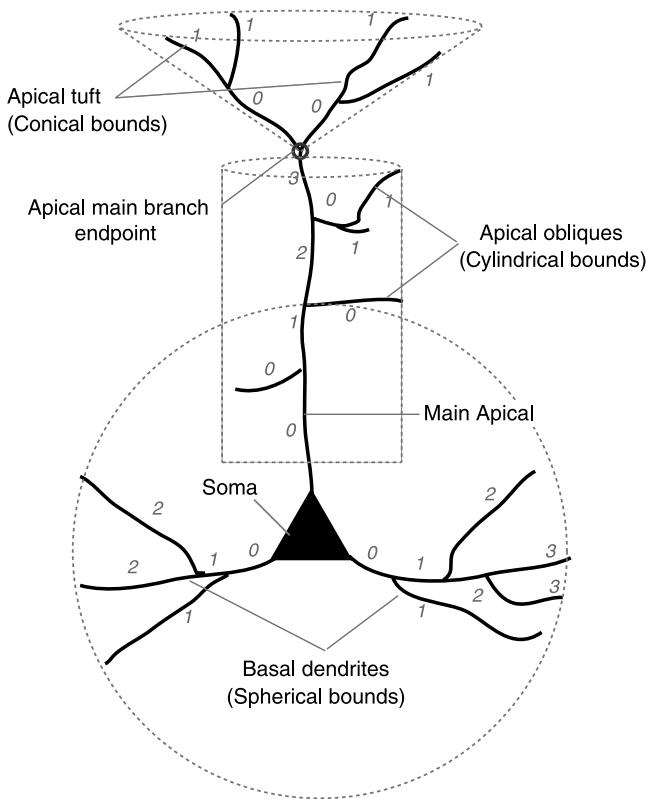
Repairing Cut Dendrites

Repairing neurons requires the identification of cut branches. It is crucial to correctly identify cut branches for two reasons: the cut branches are used to extract the statistical features, and they are the starting points for the repair. Cut points are identified based on the assumption that all cuts occur in a single two-dimensional plane, owing to the preparation of the *in vitro* slice. We further assume that the slices have been placed under the microscope so that the cut plane is parallel to the XY plane. Based on this assumption, we define an additional XY plane aligned with the soma center, thereby delineating two regions in which cut planes could occur. The algorithm attempts to identify the region with the cut plane by comparing the branch ending distances with their distribution. The cut plane occurs in the region where the algorithm finds the greatest number of branch terminals with a terminal distance less than the (mean – standard deviation) of the branches' ending distance. All points lying in this region are identified as the cut branch terminals of the neuron, as shown in figure 9.2. Additional cut branches can be manually specified.

The algorithm for the repair of neurons exploits the assumption of a statistical symmetry to compute a probability density cloud for the uncut dendrites of the neuron and then uses the resulting probability density functions (representing the probability density cloud) to grow the cut dendrites of the same neuron. The repair algorithm is described in detail in the following sections.

Selection of Apical Tuft

Most of the neurons with apical dendrites need further classification of the apical dendrites into the main, oblique, and tuft branches. This classification serves to define statistics for the repair as well as providing useful information for volumetric boundaries, as discussed later. The pseudoclassification of apical dendrite segments into branches requires the correct classification of the end point of the apical main branch. However, there is significant variability in the structure of apical branches; therefore the end point of the apical main branch is identified manually. After the selection of this end point, portions of the apical branch are classified as belonging to the main apical branch, apical obliques, or apical tufts. The main apical branch contains all the children of the first segment starting from the soma, where the end point of each child segment forms the starting point of the next child, until it reaches the end point of the apical main branch. The apical tuft includes all segments whose parents branch at the end point of the apical main branch. The apical oblique branches are formed by the segments whose parents originate from the main apical branch, except for those from the apical main branch end point.

**Figure 9.4**

Definitions used in the repair of cut branches. The branch order of the segments is shown after pseudo-classification of apical branches, together with the boundary conditions for the different classes of dendrite segments.

The pseudo-orders for apical dendrites (as in figure 9.4) are computed after the further classification of the apical dendrite into the main branch, obliques, and tufts. The oblique branches, whose parent segments belong to the main apical branch, are considered root branches. The tuft branches, whose parent segments are part of the apical main branch, are considered as the root apical tuft branches where the pseudo-order for the apical main branch is the same as before classification.

Generate a Pool of Reconstruction Blocks

The purpose of the repair is to recover missing information about the morphologies while preserving the statistical features of the neuron. The building blocks of the repair consist of segments and bifurcations. We use the building blocks from the neuron itself to regrow the missing dendrites. In this way we aim to preserve some

structural features of neurons that are less sensitive to whether the morphology is complete (e.g., interbranch lengths and angles). The common information stored in the pool of reconstruction blocks is the type of source branch, i.e., the apical main branch, obliques, tufts or basal, and the branch order of the source (pseudobranch order in the case of apical branches). The segments preserve the starting and ending diameters as well as the length of the segment. Bifurcations are represented by three segments; one is the parent segment and two segments are children of the parent. The information required to represent a bifurcation includes the diameters and the coordinates of the four points (two points for the parent and one point for each child).

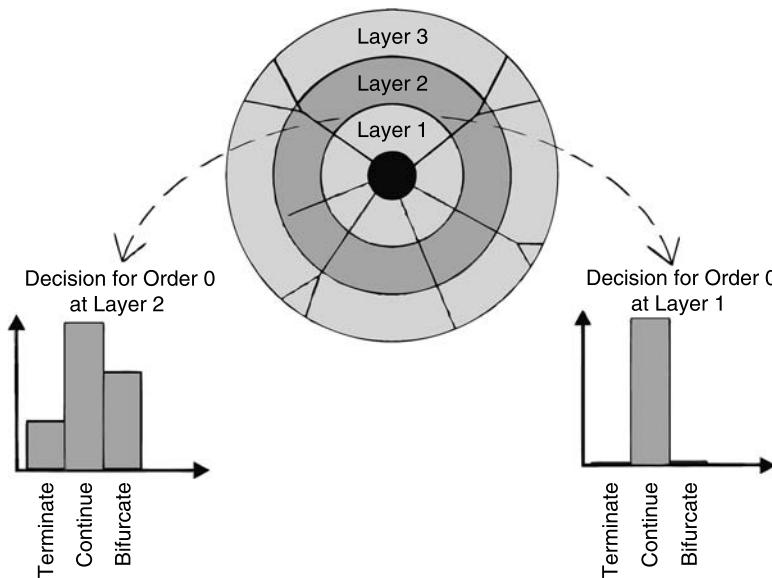
Evaluate Volumetric Bounds

Beyond the statistical characterization of morphological features, each dendritic portion of a neuron occupies a distinct volume. These volumes are complex but can be represented approximately by simple shapes like a sphere, cylinder, or cone. After the identification of cut branches, we need to know within which bounds a neuron should be regrown. The volumetric bounds, therefore, are essential to avoid an unusual shape of the neuron, and they depend on the type of neuron structure.

For the basal dendrites of pyramidal cells, the radius of a spherical bound is defined by the maximum of the distances computed for the uncut basal dendritic branch terminals from the soma center. For the apical obliques, a cylindrical bound is placed vertically at the center of the main apical branch. The radius of the cylinder is derived from the maximum perpendicular distance from the uncut oblique terminals to the main apical branch. The height of the cylinder is determined from the first oblique branch point and the last oblique branch point. For the apical tuft, we use a cone with its tip placed vertically at the main apical branch end and its axis aligned with the main apical branch. The height of cone is computed as the distance from the projections of uncut tuft terminals to the main apical branch end point of the main apical branch. The angle of the cone is calculated by finding the maximum of the angles formed by the uncut terminal tuft branches.

Evaluation of the Probability Density Cloud

The dendritic space is divided into regions using concentric spheres with variable radii (e.g., 10- μm intervals) placed around the soma center, as shown in figure 9.5. The statistics of different behaviors (branching, terminating, and continuing) are computed for each branch order and distance range within shell S (equation 9.1). When the probability density cloud is computed, certain branch orders and segment distances will result in a probability of continuation of zero and a probability of termination of one. These statistical properties therefore determine the bounds of the growth and implicitly capture the appropriate boundary conditions for branch order and segment distance. Evaluating the statistics within volumes of appropriate shapes could eliminate this requirement.

**Figure 9.5**

Schematic showing the formation of the probability density function. This is based on branch properties at subsequent branch orders and Sholl distance.

Repair Algorithm

The repair algorithm for dendrites starts with the identification of cut branches as described earlier (figure 9.6, left). Only these identified cut branches are repaired to restore the structure of the neuron (figure 9.6, right). These branches are regrown under the assumption of an underlying statistical symmetry in the neuron dendrites. Implicitly, these statistics provide a limit on the growth of the repair because the probability of termination for each segment ultimately becomes one. Furthermore, the growth process is constrained by volumetric boundary conditions, as described earlier. The repair process restores the assumed symmetry of the neuron (figure 9.7). The probability of continuation, bifurcation, and termination is evaluated for the valid uncut branches as shown in figure 9.8 (left). These probabilities were maintained while regrowing the cut branches as shown in figure 9.8 (right).

The repair algorithm traverses the region between each of the consecutive shells of the probability cloud in ascending order and recursively finds the cut terminals in each region. The location of the region and the order for each of the locally identified cut terminals are used to get the appropriate behavior function (probability density function). Each cut terminal can have one chance to continue, bifurcate, or terminate in one intersurface region.

The continuation of a branch consists of selecting a segment from the pool of valid segments from the intact neuron. This selection identifies the segment that best

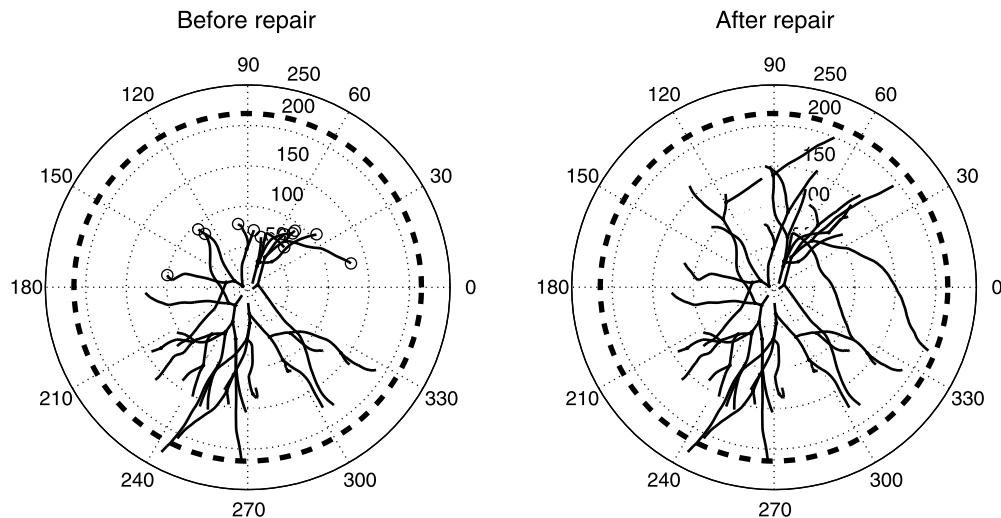
**Figure 9.6**

Left view of the pyramidal cell before (left) and after (right) dendritic repair. The identified cut terminals are marked with black circles in the left figure. The dendrites are shown as black lines and the axons in gray.

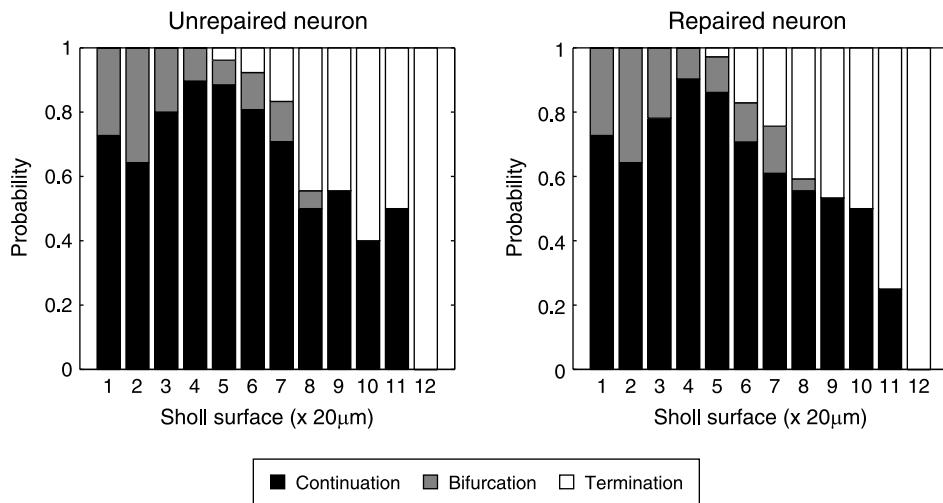
matches the diameter, order, and type of the source branch. Once the segment is chosen, the branch is grown with a small random variability (5–10%) in the direction of its parent segment.

Similarly, the bifurcation of a branch is generated by selecting from the pool of possible bifurcations. This selection identifies the bifurcation that best matches the diameter, order, and type of the source branch. Once an appropriate bifurcation is chosen, it is added to the terminal segment of the previous branch.

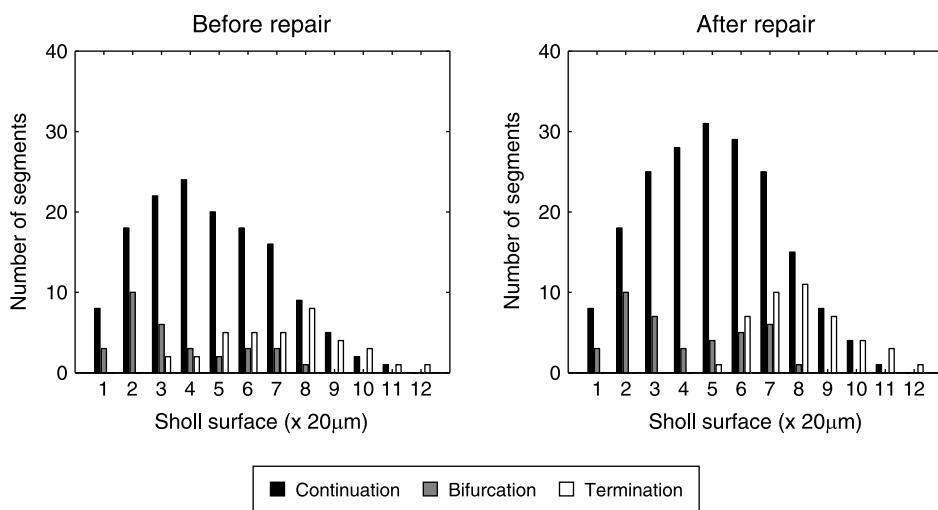
The selection of a terminating branch segment excludes the current point and cut branch from the pool of candidate cut points and cut branches. The continuation of a cut branch generates a new candidate cut point, and the bifurcation of a cut branch generates two new candidate branches. The repair of dendrites continues until there are no more cut points or cut branches left, or the growth reaches the volumetric bounds.

**Figure 9.7**

Top view of the neuron before and after repair. The dendritic repair process restores the statistical symmetry within identified volumetric bounds. The dotted line depicts the boundary conditions for the repair process.

**Figure 9.8**

The probability of branch continuation, bifurcation, and termination with Sholl distance in the neuron. The unrepaired neuron statistics (left) are based on the portion of the neuron that did not include any cut branches. The repaired neuron statistics (right) are based on the entire repaired neuron.

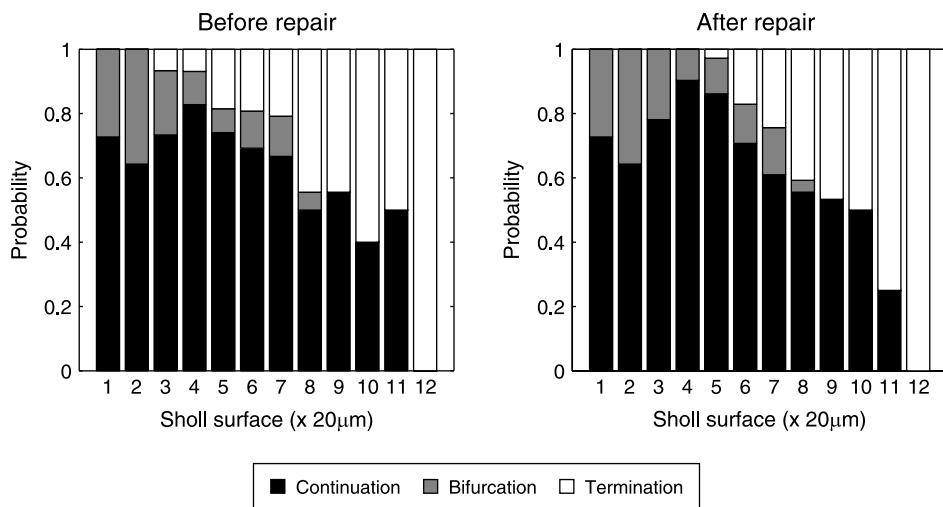
**Figure 9.9**

Count of characteristics of branches of a neuron before and after dendritic repair as a function of distance from the soma center. Continuing branches are black, bifurcating ones are gray, and terminating branches are white. The neuron before repair has many branches terminating at distances of 60–100 μm, which were detected as the cut branches. Segments and bifurcation replace these cut branches. Overall, the general shape of each distribution is preserved.

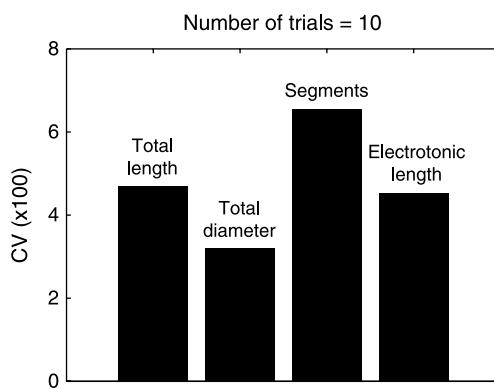
During repair, terminations that have been identified as cut points are replaced by new segments, which are selected using the probabilities determined from the intact neuron (figure 9.8). The replacement of “false” terminations with continuations and bifurcations changes the distribution of segments at different spatial distances, resulting in an increase in the number of segments in the repaired neuron (figure 9.9). This increase in segments also influences the total and electrotonic lengths; thus the repair can have an important role in restoring the electrical properties of the neuron.

The probabilities of termination are relatively high when analyzing a cut, unrepairs neuron because of the inclusion of all the terminations that are due to cutting. The repair process restores the probabilities of the entire neuron to match those of the intact portion of the neuron (compare figures 9.8 and 9.10). This change in the probabilities demonstrates the success of the process in repairing a neuron based on its intact portion.

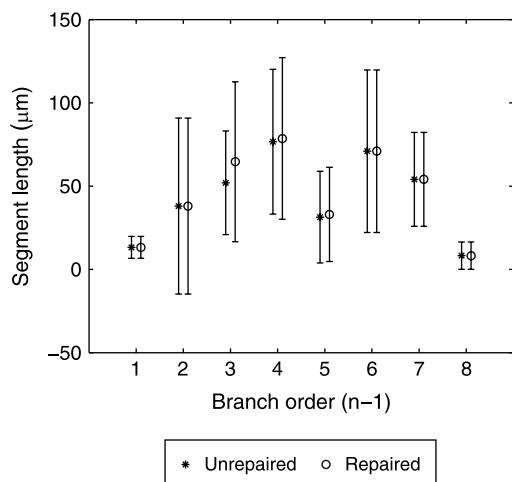
The repair process results from the evaluation of multiple joint probability functions and produces a different repaired neuron with each new random number seed. To assess the consistency of single-cell properties with the repair, we evaluated the length, diameter, number of segments, and electrotonic length for ten trials of repairing a single cell. Each repaired neuron varied by less than 8% in each of these single-cell measures (figure 9.11). The repair, while producing completely new morphologies, maintains important single-cell morphological and electrical properties.

**Figure 9.10**

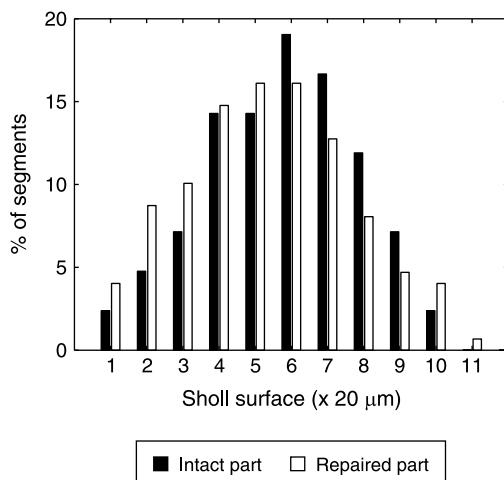
The probability of branch continuation, bifurcation, and termination with Sholl distance in the neuron. These statistics were computed by including identified cut branches, before (left) and after (right) repair. The characteristics of the neuron before repair illustrate the elevated termination probabilities at a 60–100-μm distance in-between the third and fifth probability shells that is due to the cut branches. These characteristics change after a repair of the neuron that replaces terminations with the continuations and bifurcations in these regions.

**Figure 9.11**

Coefficient of variability (CV) of the repair process. The repair process produced neurons with an overall variability of less than 8% in dendritic characteristics that included the total length, diameter, number of segments, and electrotonic length.

**Figure 9.12**

The segment length distribution of dendritic branches as a function of branch order. Mean \pm standard deviation before the repair (excluding the cut branches) and after the repair are shown.

**Figure 9.13**

Percentage of segments across Sholl distance. Two distributions are shown: one for the intact portion of the dendrite (excluding all subtrees with cut branches) of the neuron and the other for the repaired part of the neuron.

Validation

We used a layer 2 pyramidal cell to validate the dendritic repair process as described in the previous section. The features examined were the segment lengths and the number of segments found per Sholl shell. These are features that were not explicitly modeled in the repair process.

The segment lengths were sampled from the unrepairs neuron, excluding cut branches, and compared within their branch order against the segment lengths obtained from the repaired neuron. The mean segment lengths of the repaired branches were found to be not significantly different from the segment lengths of the unrepairs neuron, as shown in figure 9.12.

The symmetry of the neuron can be examined by the distribution of branches in two regions separated by the soma—one region with intact branches and another with cut branches. Asymmetries in the distribution of segments across the Sholl distance can reveal the impact of the cuts on the dendritic morphology. As expected, the repair process improves the symmetry of the distribution of segments over Sholl distances (figure 9.13).

9.3 Repair of Neuronal Axons

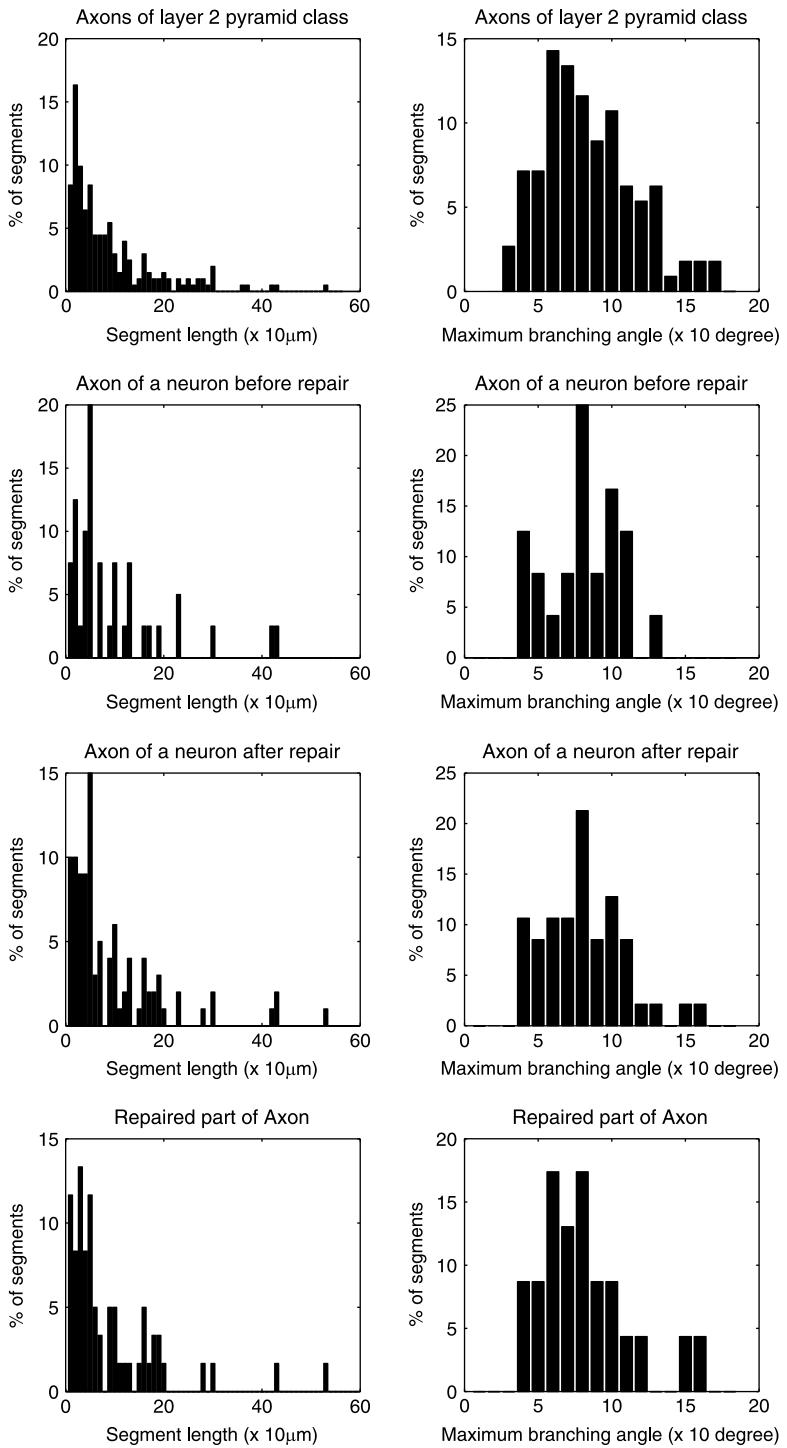
Axons do not exhibit the same symmetry as dendrites and therefore their reconstruction cannot be based on the assumption of statistical symmetry. The axon repair starts with the analysis of global morphometric statistics for an entire morphological class. The building blocks are the intact subtrees taken from each existing axon branch order across the morphological class. The morphometric parameters for each subtree are evaluated for use in the axon repair. The aim of the repair is for the regrown part to match the statistics of the intact axon.

Evaluation of Morphometric Parameters for Axons of the Same Neuronal Class

The morphometric parameters are computed on the intact subtrees (with no cut branches) from the class of neurons used for the repair. These statistics include axonal segment length, axonal branch order, planar axonal branch angle, and maximum axonal branch angle, as shown in figure 9.14.

Evaluation of Probability Functions to Grow Axonal Trees at Cut Points

The probability functions are derived as the difference between morphometric parameters of the intact axonal subtrees of neurons belonging to the anatomical class and morphometric parameters of the portion to be regrown (as shown in figure 9.14). The statistics of the portion to be regrown would start as empty and will incrementally be updated with each subtree that is pasted. These functions are used in the following way:



1. The probability density function representing the axonal segment length indicates the lack of segments with particular lengths.
2. The probability density function representing the axonal branch order indicates the lack of branches of a particular order.
3. The probability density functions representing the maximum axonal branching angle and planar branching angle indicate a lack of the particular angle bifurcations.

Therefore a comparison between individual morphometric parameters of the population and the cut part that is being regrown acts as a guide in repairing the axon.

Repair Algorithm

The repair of axon cut terminals is designed to preserve the specific morphometric parameters listed above. All cut terminals are analyzed one-by-one for the repair, and a pool of intact subtrees with the order of the current cut terminal is extracted for each of the cut branches from the database of subtrees. The morphometric parameters for the chosen pool of subtrees are also extracted. The subtree to be appended at each cut terminal is selected on best-fit criteria. The best-fit criterion in our algorithm is defined as the maximum compensation of lacking morphometric parameters of the part to be regrown compared with the intact subtrees' morphometric parameters for the population of the same anatomical class. Correlations between the probability functions related to morphometric parameters and the morphometric parameters of each of the subtrees are computed and the subtree with the maximum correlation is appended. After the repair of the axon, the morphometric parameters of the regrown part should capture the average morphometric specifications of intact axons belonging to its anatomical class.

Validation of Axon Repair

The axon repair succeeds in restoring the statistics of an individual cell according to a population of cells. However, additional data for neurons with intact axons are required to validate the axonal repair. These data are extremely rare from *in vitro* preparations. Therefore the repair process should ultimately be validated by using an intact neuron reconstruction (such as that from a neuron recorded and filled *in vivo*). By simulating a cut through an intact morphology and then repairing the miss-

Figure 9.14

The morphometric parameters of layer 2 pyramidal cell axons. The left column shows the axonal segment length, i.e. the length of axonal branches (excluding cuts). The right column shows the maximum axonal branching angle, which is the maximum of all angles that child branches form with their parent branch. The top two panels show the morphometric statistics for the entire morphological class; the other panels show statistics for a single neuron before and after repair and for the repaired part exclusively (bottom panels).

ing portion, the success of the repair process can be evaluated. A measure of the success of the repair will be a statistical comparison of the restored portion with the original portion removed by the simulated cut. Special consideration will have to be made to account for long-range and patch axonal collaterals. Future work will address these issues.

9.4 Conclusions

The routine reconstruction of neuronal morphologies from *in vivo* preparations is currently too complex and therefore provides too sparse a selection of neuron morphologies to be adequate for modeling cortical microcircuitry. The alternative, which is to use reconstructed morphologies from *in vitro* experiments, allows cells to be visually preselected and systematically recorded. The tradeoff of this approach is the loss of morphological accuracy caused by potentially incomplete arbors of the neurons that are due to preparation of the slice. We have presented an algorithm based on statistical symmetry assumptions about the morphologies of cortical neurons that used the available characteristics of the neuron to recover the missing structures. In this chapter, we focused on the underlying principles and presented an analysis that demonstrates the successful application of the algorithm to the repair of a layer 2 pyramidal cell.

The primary assumption in our repair algorithm is that the dendrites of neurons have a statistical symmetry in their morphologies, although they do not necessarily appear geometrically symmetric. We assume that the statistical behavior of the missing portions (i.e., the probability of bifurcation, termination, and continuation in a region defined by the probability cloud), should be approximately the same as that of the intact parts in that region. This provides a guess about the missing parts of the dendrites in that region based on the information from intact parts. The repair will not necessarily force the dendrites to grow symmetrically and reach the maximum bound; this decision rests with the probability of continuations. Ultimately, however, it may give rise to a geometrical symmetry at the level of the complete dendritic arbor if the symmetry exists at the level of individual dendritic trees and subtrees.

Before a neuron is repaired, it is unraveled to reverse the shrinkage that is due to the histological preparation. To accurately estimate shrinkage, landmarks are often created during the experiment by making regularly spaced parallel penetrations in the region of interest. The shift in these locations determines the shrinkage factor. The window length of the unraveling process influences the amount of expansion and can be selected to best restore the original locations of landmarks. Careful measurement, analysis, and correction of the tissue shrinkage will improve the repair results.

The identification of cut branches was based on the assumption that all the branches in a neuron reach the bounds of the neuron. This assumption may not always hold true, particularly in morphologies where the cells are completely asymmetrical. In that case, careful manual identification of the cut branches would be essential to ensure that only these branches are grown.

The boundary conditions of the neuron were computed after classification of branches into basal dendrites, apical obliques, and apical tuft, which is the case in most pyramidal cells. This classification may need to be extended because some pyramidal cells have two tufts and sometimes the tuft is so extensive that it needs careful handling in the repair.

The spatial characteristics of the uncut branches were used to repair the cell, which assumes the neuron has an underlying statistical symmetry. In the case of cells with an asymmetry, it may be difficult to repair the cell, owing to the lack of spatial and statistical characteristics as well as a lack of information about the volumetric bounds. In that case, we propose using the average probability density cloud obtained from an entire class of neurons to complete the dendritic repair process, similar to the axonal repair process. This technique could also be used for partially stained cells if they are well classified. However, this technique would increase the dependence of the repair on the quality and quantity of available cells belonging to the same anatomical class of neurons.

An aspect of neuron morphology that has been neglected in this chapter is the importance of dendritic spines (chapter 8, section 2). Dendritic spines play a key role in synaptic connectivity, and their distribution and length will significantly affect the connectivity between neurons in a network. Future work should include information on synaptic spine density in the repair process to address this issue.

The repair of axons was based on morphometric parameters obtained from a population of neurons belonging to the same anatomical class. Therefore the algorithm depends on the quality as well as the quantity of the population of cells. Further studies are required to document precisely how many neurons are needed to obtain sufficient accuracy in the statistical data to yield a measurable degree of accuracy. In vivo studies will also be required in order to obtain more accurate information on the long-range patchy axon collaterals of pyramidal cells.

10 Passive Cable Modeling

William R. Holmes

The application of passive cable theory to neuronal processes has a long and rich history (reviewed by Rall, 1977). Neuronal processes, being long and thin with an electrically conducting core surrounded by a membrane with high resistance, are naturally described as core conductors. Cable theory (named for its application to the first transatlantic telegraph cable) provides a straightforward means to study core conductors and has proved to be highly relevant and useful for the study of neuronal processes. Passive cable theory assumes that membrane properties are constant (independent of voltage and time). Although there is much evidence that membrane properties are not passive (see chapter 11), passive cable theory and passive cable modeling remain important for several reasons. First, they provide an intuition that is difficult to attain otherwise for understanding how neurons function; nonlinear properties are notoriously nonintuitive. Second, passive cable theory provides a number of significant insights whose usefulness carries over to the nonlinear case. Third, passive neuron models provide an important starting point or reference case from which one can build more complex models with excitable properties. Fourth, the propagation and summation of synaptic inputs are largely determined by passive properties. Finally, the effects that voltage-dependent conductances have on dendrites are heavily influenced by the passive electrotonic structure.

This chapter introduces basic passive cable theory. It describes properties of passive systems important for experimentalists and presents some insights from cable theory, many of which have received too little attention. It then discusses from a historical view how to estimate electrotonic parameter values from experimental data, along with problems associated with these older techniques. Finally, modern methods for estimating parameter values from data for passive systems and guidelines for constructing passive models are discussed, along with potential pitfalls that one should know. More detailed expositions and additional background on many of these topics are given in monographs by Jack et al. (1975), Koch (1999), Rall (1977), Rall et al. (1992), Rall and Agmon-Snir (1998), and Segev et al. (1994).

10.1 Electrotonic Parameters, the Cable Equation, and the Equivalent-Cylinder Model

To understand insights from cable theory, it is useful to review briefly some basic properties of neurons, show where the cable equation comes from, discuss some misconceptions about the meaning of some electrotonic parameters, and introduce the equivalent-cylinder model.

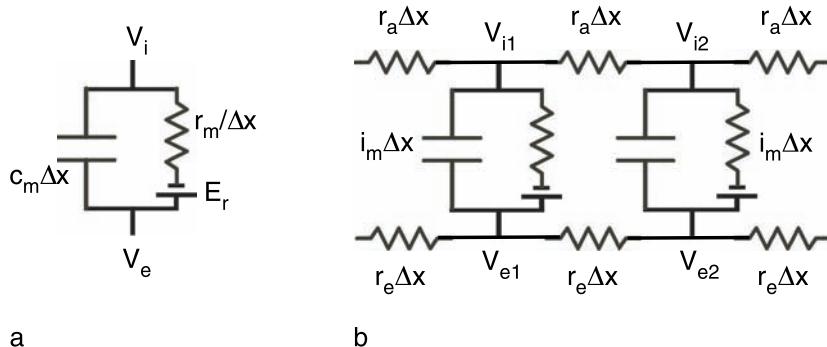
Membrane Potential and Input Resistance

There is a voltage difference across the neuron membrane resulting from the different distributions and membrane permeabilities of ions and charged molecules on either side of the membrane. This voltage difference across the membrane typically ranges from 40 to 90 mV among cell types, with the inside being negative relative to the outside. We say that cells have a resting membrane potential, V_m , of -40 to -90 mV. If we inject constant current into the cell, the voltage will change to a new steady-state value, illustrating another basic property of neurons, the input resistance, R_N . For passive systems, input resistance is computed simply from Ohm's law as $R_N = \Delta V / \Delta I$ where ΔV is the change in voltage that occurs with the change in applied current, ΔI . Input resistance is an important parameter to measure because it illustrates in a general way how excitable a cell is in response to synaptic input.

Input resistance, R_N , as a characteristic property of a cell, is measured at the soma. However, it is useful at times to refer to a local input resistance. This is the input resistance that would be measured at a dendritic location if an electrode could be placed there. Because dendritic processes are often very thin, local input resistance is typically larger than the input resistance measured at the soma (see section 10.2), but this is not necessarily the case.

Membrane Capacitance and Resistance

When a current is injected into a cell, the new steady-state voltage is not reached instantaneously. This can be explained by considering the properties of the membrane. The neuron membrane is a lipid bilayer of limited thickness and because the voltage inside is negative relative to the outside, the membrane stores and separates charges. We say the membrane has a capacitance. The amount of charge, Q , separated by a capacitor is given by the formula $Q = CV$ where C is the characteristic capacitance of the membrane and V is the voltage difference across the membrane. The specific membrane capacitance, C_m , for neurons has long been considered to be a biological constant (Cole, 1968) with a value of approximately $1.0 \mu\text{F}/\text{cm}^2$. This value is based on measurements made from the squid axon (Curtis and Cole, 1938) and may be a slight overestimate. Squid axon has a high density of proteins that form voltage-dependent ion channels in the membrane. Artificial lipid bilayers with no

**Figure 10.1**

Conceptual models for neuronal membrane. (a) Simple circuit for a patch of membrane. (b) Circuit for a neuron cable. V_i is the intracellular voltage, V_e is the extracellular voltage, E_r is the membrane battery, r_m is the resistivity of a unit length of membrane (Ωcm), c_m is membrane capacitance per unit length ($\mu\text{F}/\text{cm}$), r_a (also called r_t) is axial resistivity per unit length of cable (Ω/cm), r_e is extracellular resistivity per unit length (Ω/cm) and Δx is the length of the patch. These are related to the specific units of measure described in the text by $r_m = R_m/\pi d$, $c_m = C_m\pi d$, $r_a = 4R_a/\pi d^2$, where d is the diameter of the cable.

embedded membrane proteins tend to have a lower C_m (Benz et al., 1975; Niles et al., 1988), with values ranging from 0.65 to 0.94 $\mu\text{F}/\text{cm}^2$, depending on the lipid composition and the thickness of the hydrocarbon core. More recently, Gentet et al. (2000) made direct measurements of C_m in cortical pyramidal, spinal cord, and hippocampal neurons and reported an average value of 0.9 $\mu\text{F}/\text{cm}^2$.

The lipid bilayer also provides a resistance to current flow. If the membrane were a pure lipid bilayer, the specific membrane resistivity, R_m , would be in the neighborhood of $10^8 \Omega\text{cm}^2$ (Almers, 1978) or higher (Hille, 2001). However, the membrane is studded with proteins, some of which form ion channels, and this reduces the membrane resistivity at rest to the 10^4 – $10^5 \Omega\text{cm}^2$ range. Given that the membrane has a capacitance and a resistance, it is natural to develop a conceptual model of the membrane as an electrical circuit as in figure 10.1a, where the battery, E_r , represents the membrane potential, and to consider membrane current as the sum of the current flows through the separate capacitive and resistive branches of this circuit.

Axial and Extracellular Resistance

The simple conceptual model of figure 10.1a applies for a patch of membrane or for spherical or isopotential cells, but neuronal processes more typically resemble cylinders or cables. In neuronal cables, current flows not only through the capacitance and resistance of the membrane, but also through the interior of the cell and in the extracellular space as well. When these factors are considered, our conceptual model of the neuron cable becomes the electric circuit shown in figure 10.1b. Strictly speaking, there is three-dimensional current flow inside the neuronal cable, but in practice

we consider only the longitudinal current down the cable through the axial resistance provided by the intracellular medium. This is reasonable because the process diameter is generally small, making intracellular resistance to radial current flow negligible compared with the membrane resistance; similarly, we also neglect intracellular resistance to angular current flow.

The specific intracellular or axial resistivity, R_a , is a difficult parameter to measure (see discussion in Rall et al., 1992). Intracellular resistivity is traditionally labeled R_i . Unfortunately some papers incorrectly use R_i to mean input resistance. To remove all ambiguity, we use R_a for intracellular or axial resistivity and R_N for input resistance. Estimates for R_a range from 50 to 400 Ωcm in mammalian neurons; lower estimates have been obtained in marine invertebrates where the intracellular ion concentrations are much different. For example, the resistivity of “Woods Hole seawater” is 20 Ωcm and the resistivity of squid axoplasm, measured relative to seawater, is 1.0–1.6 times higher (Cole, 1975; Carpenter et al., 1975). By comparison, at 20°C, mammalian saline has an R_a of 60 Ωcm and frog Ringer’s solution has an R_a of 80 Ωcm (Hille, 2001). Barrett and Crill (1974) measured an average R_a of 70 Ωcm in cat motoneurons at 37°C. Values of 100–250 Ωcm have been reported for red blood cells and frog skeletal muscle (Pilwat and Zimmerman, 1985; Schanne and De Ceretti, 1971). Clements and Redman (1989) computed a value of 43 Ωcm at 37°C for motoneuron cytoplasm based on ion composition, but noted that the actual R_a will be higher than expected from known concentrations of ions and ion mobility because of the presence of charge binding and various proteins, carbohydrates, and organelles. Different intracellular compositions may explain the variability in measured values for the same cell type as well as that among different cell types. Other factors affecting R_a are the composition of the extracellular medium (Schanne, 1969; the mechanism is unclear; it may be because of an induced change in the free and bound concentrations of ions inside) and temperature (R_a will decrease with increasing temperature; Trevelyan and Jack, 2002).

Extracellular resistance is usually assumed to be negligible in models. While this is clearly not true (because extracellular recordings could not be made otherwise and ephaptic coupling has been observed among tightly packed neurons; Jefferys, 1995), it is a reasonable assumption in most situations when the extracellular space is large (e.g., see Rall, 1959), the neuron is isolated, or the concern of the model is a single neuron.

Converting the Conceptual Model to the Cable Equation

The conceptual models represented in figure 10.1 can be converted into mathematical equations by straightforward application of Ohm’s law, $Q = CV$, and Kirchoff’s law (conservation of current at a node). The most difficult part of this conversion is to keep track of the units of the various parameters. The mathematical equations for the conceptual models in figure 10.1 are

$$i_m = c_m \frac{dV}{dt} + \frac{V}{r_m}, \quad (10.1)$$

which says the membrane current equals the sum of the capacitive and resistive currents across the membrane and

$$\frac{1}{r_a} \frac{\partial^2 V}{\partial x^2} = c_m \frac{\partial V}{\partial t} + \frac{V}{r_m}, \quad (10.2)$$

which says the change (or difference) in intracellular current flowing along the inside of the cable equals the current that flows across the membrane. Equation (10.2) is usually multiplied by r_m and rearranged to yield the cable equation

$$\tau \frac{\partial V}{\partial t} = \lambda^2 \frac{\partial^2 V}{\partial x^2} - V, \quad (10.3)$$

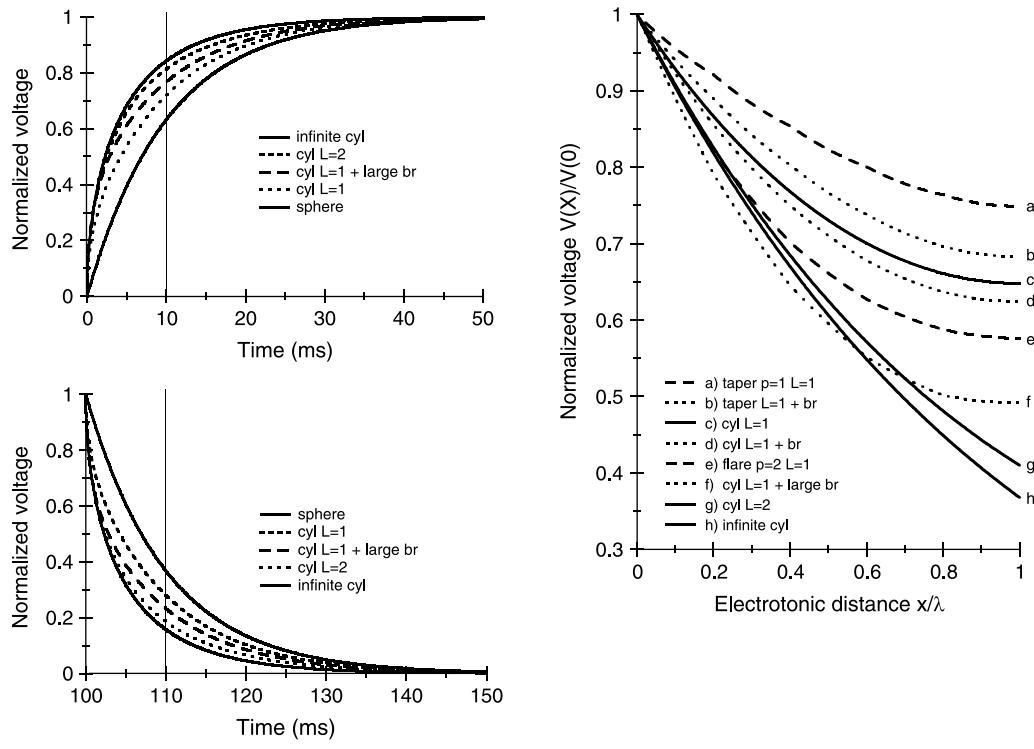
where τ is the membrane time constant and λ is the space constant defined by

$$\tau = r_m c_m = R_m C_m \quad \lambda = \sqrt{\frac{r_m}{r_a}} = \sqrt{\frac{R_m d}{4 R_a}}. \quad (10.4)$$

(See figure 10.1 caption for notation definitions.) Note that if extracellular resistance is included in the derivation, then $r_a + r_e$ is substituted for r_a in the definition of λ . Equation (10.3) is often expressed in nondimensional form by defining $T = t/\tau$ and $X = x/\lambda$ to get rid of those pesky constants and allow a more straightforward mathematical analysis. Other useful terms are L , the electrotonic length, which is defined by $L = \ell/\lambda$ where ℓ is the physical length of the cable; and the electrotonic distance X , defined as earlier by $X = x/\lambda$, the physical distance from the beginning of the cable divided by the space constant. The parameter L will appear numerous times in the following discussion.

Misconceptions about τ and λ

Equation (10.1) and the steady-state version of equation (10.3) (left-hand side = 0) are easily solved and the solutions are often used in textbooks to indicate the importance of τ and λ for electrical signaling. However, this is often done in a misleading way. A solution for equation (10.1) is $V(t) = V_0 e^{-t/\tau}$, from which it is usually concluded that τ is the time it takes for a voltage perturbation to decay from rest to 37% ($1/e$) of its initial value. However, rarely is it stressed that this is true only for an isopotential cell or sphere (the conditions leading to equation 10.1). For cylinders, the decay (as well as charging) is much faster (figure 10.2a). In an infinite cylinder, voltage will be only 16% of its initial value at time $t = \tau$, and in a finite cable with an electrotonic length $L = 1.0$, voltage will decay to 28% of its initial value in this

**a****b****Figure 10.2**

Effect of geometry and boundary conditions on transient and steady-state voltage changes. (a) A constant current is applied at time $t = 0$ ms for 100 ms. The upper curves show the voltage during the first 50 ms of the charging phase and the lower curves show the voltage decay for the first 50 ms after the current is terminated. Note the symmetry of the charging and decay transients. Curves top to bottom are in the same order as in the legends and represent the responses in an infinite cylinder, a sphere, finite cylinders with electrotonic lengths of 1 and 2, and a cylinder with an electrotonic length of 1 with a large side branch attached. The vertical lines at 10 and 110 ms are times $t = \tau$ after the current injection is turned on or off ($\tau = 10$ ms). (b) Steady-state voltage decay with distance. Geometries include cylinders with thick (f) and thin (b, d) side branches. Cases with flare (e) and taper (a) were constructed with symmetric trees with diameters of parent and daughter branches satisfying $(d_{\text{parent}})^p = \sum(d_{\text{daughters}})^p$ at branch points where $p = 2$ represents flare and $p = 1$ represents taper. Electrotonic length was 1.0 in all cases except (g) and (h).

time. In a finite cylinder, there are additional, faster equalizing time constants that increase the rate of the decay. These are described in more detail later. In figure 10.2a the charging transients are also shown. It is not readily appreciated by those new to the field that charging and decay transients are symmetric.

Similarly, a steady-state solution for equation (10.3) is $V(x) = V_0 e^{-x/\lambda}$, from which it is usually concluded that λ is the distance at which the steady-state voltage decays to 37% ($1/e$) of its value at the origin. However, rarely is it stressed that the solution and the conclusion are valid only for an infinitely long cable. For a finite cable with an electrotonic length $L = 1.0$, voltage will decay to only 65% of its value at the origin at a distance $x = \lambda$. Differences such as this arise because of the boundary conditions at $x = \lambda$. In the infinite-cylinder case, an infinite cylinder with its associated conductance is attached to the (artificial) boundary at $x = \lambda$ and current flows into this cylinder, whereas in a finite cylinder with $L = 1.0$, the cylinder ends at $x = \lambda$. For a sealed-end boundary condition, there is no conductance and no current flow at this boundary (assuming a leaky end through a small end cap of a membrane makes a negligible difference). Voltage decay with distance is also affected by taper or flare in the diameter of the equivalent cable and by side branches off the main cable. Examples of this are shown in figure 10.2b.

Nevertheless, τ remains a useful measure of voltage decay with time and λ remains a useful measure of voltage decay with distance. However, it is important to understand how boundary conditions and complex dendritic geometry can influence temporal and spatial voltage summation and voltage decay in a neuron.

Equivalent-Cylinder Model

Although neurons are highly branched structures rather than single cylinders, a highly branched neuron can be equivalent mathematically to a single cylinder when a particular set of conditions holds. This is Rall's equivalent-cylinder model (Rall 1962a,b). Starting with the most distal branches and working inward, each pair of daughter branches is replaced by a single cylinder having the same input conductance and electrotonic length as the two daughter branches but having the same diameter as the parent branch. This is possible when R_m and R_a are uniform; all terminal branches have the same distal boundary condition (e.g., no current through the end of the branch); all terminals end at the same electrotonic distance from the soma; and the daughter branch diameters raised to the $3/2$ power and summed equal the parent branch diameter raised to the $3/2$ power (often called the $3/2$ rule or $3/2$ power law). These constraints also preserve input conductance and membrane area between the branched tree and the equivalent cylinder.

The equivalent-cylinder model is a useful construct for studying the effect of current input or a voltage clamp applied at the soma on voltage changes throughout the dendritic tree. Its simplicity allows mathematical analysis and ready insight into

dendritic function. However, it is less useful for studying dendritic input. Any input to a location in the equivalent cylinder would have to be divided among all dendrites in the fully branched structure at the same electrotonic distance for the responses to be the same. The equivalent-cylinder model is also useful for estimating parameter values, as discussed later. Clearly no dendritic tree is likely to satisfy all of the constraints of the equivalent-cylinder model, particularly the constraint of all dendrites terminating at the same electrotonic distance from the soma. A number of neurons seem to follow the 3/2 rule over much of their dendritic trees (Barrett and Crill, 1974; Turner and Schwartzkroin, 1980; Durand et al., 1983) but others do not (Larkman et al., 1992).

10.2 Properties of Passive Systems

To model complex branching structures, one can apply the cable equation (equation 10.3) to every unbranched segment of the dendritic tree and then match boundary conditions (conservation of current, continuity of voltage) at the branch points. While there are straightforward ways to do this analytically for the steady-state case, the mathematics quickly becomes very messy and complicated for the transient or time-dependent solution. A simplification is to make unbranched segments small enough so that they are essentially isopotential, model the membrane current of these segments with equation (10.1), and then couple adjacent segments with “coupling resistances” based on segment axial resistances (see chapter 11, section 1). This is the compartmental modeling approach, and although it is simpler because the system of partial differential equations is reduced to a system of ordinary differential equations, the mathematics again becomes messy very quickly. It is much more convenient to model complex branching structures numerically, and there are many freely available software packages, such as NEURON and GENESIS (see the software appendix), for doing this. Because passive models only require morphology, R_m , R_a , and C_m , they are solved particularly quickly with these software programs and model results can then be compared with data from experimental cells whose voltage responses have been made passive through the use of pharmacological blockers.

Linearity (and Nonlinearity)

A key property of a passive system is its linear response to current injection. Input resistance can be calculated from any single constant current injection because Ohm’s law applies. Doubling the current will double the size of the voltage response, leaving the input resistance value unchanged. Perhaps less intuitive is the fact that if current is injected in two different places in the cell, the voltage response at any location (both transient and steady state) will equal the sum of the voltage responses to the individual current injections applied separately. However, a passive system is

not linear when it comes to synaptic inputs. Synaptic currents can be modeled as $g_{\text{syn}}(V - E_{\text{syn}})$ or a synaptic conductance times a driving force (see chapter 6, section 6.1). If g_{syn} is doubled, the voltage response is not doubled because the driving force is reduced as the voltage approaches the synaptic reversal potential E_{syn} .

Reciprocity

Passive systems exhibit a nonintuitive symmetry in voltage responses called reciprocity (see discussion in Major et al., 1993b). The voltage response at the soma to current injection at a dendritic location will equal the voltage response at this dendritic location following the same current injection at the soma. It should be noted that the voltage responses at the injection sites and the degree of voltage attenuation between the injection site and the recording site may be drastically different, but the responses at the recording sites will be identical (e.g., figure 1 of Roth and Häusser, 2001). Checking for reciprocity is a good way to determine if a dendritic tree is passive or if blockers have been successful at making it so.

Asymmetric Voltage Attenuation

One property of passive systems discovered early on is the highly asymmetric voltage attenuation from a distal dendrite to the soma compared with voltage attenuation in the reverse direction. We first consider the case where a constant current is injected at the end of a distal dendrite. Because this terminal process is thin, axial current flow is relatively small (r_a is large) and therefore the input resistance at this location is large, producing a large voltage change. However, the steady-state voltage attenuation from the point of current injection to the first branch point will be substantial. At the branch point, current can continue to travel proximally toward the soma or else travel distally in the sibling branch. Because of the sealed-end boundary condition at the end of the sibling branch and the comparatively open-end boundary condition in the proximal direction, very little current goes in the distal direction, and as a consequence there is little voltage attenuation in the distal direction. Voltage attenuation from this branch point to the next proximal branch point is again substantial, while voltage attenuation in the distal direction in the sibling branch is again small as a consequence of the sealed-end boundary condition at the end of this distal path compared with the wide-open end in the proximal direction. This pattern is repeated as the current spreads proximally (Rall and Rinzel, 1973). Conversely, if the same current is injected at the soma, the local voltage change at the soma is relatively small because of the lower input resistance there, but voltage attenuation in the distal direction is limited because the current divides among multiple branches, with each terminal branch having a sealed-end boundary condition.

Voltage attenuation from a distal site to the soma for transient inputs can be orders of magnitude more severe than that for steady-state inputs. This fact raises

the question of whether brief distal synaptic inputs can play any role in exciting a cell. Despite the significant attenuation of the peak voltage from a distal site to the soma following a transient distal current input, the time course of the voltage response is much elongated at the soma, and the attenuation of the area of the voltage waveform (integral of the voltage over time) will only be as severe as the steady-state voltage attenuation discussed earlier. This is true regardless of the time course of the transient input in a passive system. Numerical examples of this are given in Rall and Rinzel (1973) and Rinzel and Rall (1974) for dendritic trees equivalent to a cylinder and in London et al. (1999) when R_m is nonuniform. More recent examples applied to morphological reconstructions of cortical and hippocampal neurons are given by Zador et al. (1995) and Carnevale et al. (1997). These studies use the log of voltage attenuation as the basis for a *morphoelectrotonic transform* that allows one to visualize voltage attenuation between any point in a dendritic tree and the soma (or attenuation in the reverse direction) to gain an intuitive appreciation of the functional structure of the cell.

10.3 Insights from Passive Cable Theory

Consequences for Voltage Clamp

The voltage attenuation properties of passive systems have particular significance for the extent of a space clamp in the dendrites when the soma is voltage clamped. Although voltage attenuation from the soma to the distal tips is much less than the voltage attenuation in the reverse direction, the amount of attenuation is still significant. For example, suppose a cell with resting potential of -70 mV is approximated as an equivalent cylinder with electrotonic length $L = 1.0$ and the soma end is voltage clamped to -10 mV. Then the distal tips would be clamped to $-70 + 60/\cosh(L) = -31$ mV. Passive cable theory allows one to estimate the space-clamp error for a given voltage clamp at the soma, and suggests that this error can be quite significant (Johnston and Brown, 1983; Rall and Segev, 1985; Spruston et al., 1993). One consequence of this is that synaptic reversal potentials can be estimated accurately in a voltage-clamp experiment only when the synapse is very close to the soma (Johnston and Brown, 1983; but see Hausser and Roth, 1997b). Suppose one is able to insert a voltage-clamp electrode into a dendrite. Because voltage attenuation from a distal branch toward the soma is much more severe than in the reverse direction, the extent of the space clamp in the dendrite would be extremely limited.

Consequences for the Placement of Inhibition and Excitation

The asymmetry of voltage attenuation also provides insight into why inhibition is typically located at the soma whereas excitation is typically located more distally.

Inhibition is very effective when placed at the soma because its effect will be felt throughout the cell without severe attenuation. Inhibition in distal dendrites is effective for shunting local excitation, but its effect is limited spatially (Rall, 1964, 1967; Koch et al., 1983; Segev and Parnas, 1983). As for excitation, excitatory input at the soma will not produce the large local voltage changes that occur with distal input. Although large local voltage changes with distal input will attenuate severely along the path to the soma in a passive system, they are apt to activate voltage-gated conductances in a nonpassive system. This suggests that perhaps excitation is better placed in the dendrites than at the soma, where local voltage changes would be smaller and less able to activate voltage-dependent sodium or calcium conductances.

Dendritic Filtering of Synaptic Inputs

Electrotonic parameters affect the amplitude and time course of the voltage response to synaptic inputs at the input location, but how important individual parameters are depends on whether the synaptic currents are brief or prolonged. If a synaptic input is brief, the rise time and perhaps the initial decay of the local voltage response are determined primarily by the time course of the synaptic conductance change, with the peak amplitude influenced by C_m , R_a , and the dendritic diameter, but not by R_m (Major et al., 1993a). The synaptic current will initially charge the local capacitance, start to flow longitudinally through the axial resistance, and only then will it begin to flow significantly through the membrane resistance. When the synaptic current ends, the initial local voltage decay is much faster than would be predicted from the membrane time constant. Only the late portion of the voltage decay is governed by τ_m . If synaptic currents are more prolonged, the peak local voltage response is determined by the local input resistance, which depends on R_m and R_a , but not C_m .

In passive systems, the dendrites provide a low-pass filtering of the voltage response to synaptic currents and here R_m plays an important role. Membrane resistivity will affect voltage attenuation through its influence on the space constant λ and will affect the time course of the voltage decay through its influence on τ . As the signal proceeds toward the soma, the voltage response is attenuated and becomes spread out. The time to peak and half-width of the voltage response at the soma can be used to predict how far a synapse is from the soma, with more distal inputs having a longer time to peak and a longer half-width (Rall, 1967). However, both time to peak and half-width are influenced by the synaptic conductance time course, electrotonic distance, the prevalence of side branches along the path to the soma and whether inputs are restricted to a single location; these factors make predictions of synaptic location with this method less precise (Iansek and Redman, 1973; Johnston and Brown, 1983; Redman and Walmsley, 1983; Bekkers and Stevens, 1996; Mennerick et al., 1995).

Propagation Velocity

Although mathematical expressions have been derived with various assumptions to describe the propagation velocity of a signal (Jack et al., 1975; Agmon-Snir and Segev, 1993), an intuitive way to think about propagation velocity is simply to consider velocity as distance divided by time where distance is represented by the space constant λ and time is represented by the time constant τ . If we do this, we find that velocity is proportional to $[d/(R_a R_m C_m^2)]^{1/2}$ and this expression clearly illustrates the dependence of velocity on electrotonic parameters and diameter.

Optimal Diameter

Can a neuron optimize the effectiveness of distal synaptic inputs by adjusting its morphological or electrotonic properties? The answer, perhaps surprisingly, is yes. Suppose a cell is represented by a cylinder 1,000 μm long. If current is injected into one end of the cylinder, can we choose electrotonic parameters to maximize the voltage at the other end? Clearly, voltage could be maximized if $R_m \rightarrow \infty$ or $R_a \rightarrow 0$, but what if R_m and R_a are fixed to finite and nonzero values? Then it turns out that voltage at the other end of the cylinder can be maximized for a particular choice of diameter.

The reason that an optimal diameter exists is that the effectiveness of an input will depend on its amplitude at the input site and the voltage attenuation to the soma. For a fixed length cable, if the diameter is large, then input resistance is low; an input will produce a small voltage response that will attenuate very little down the cable. If diameter is small, then input resistance is large; an input will produce a large voltage response that will attenuate considerably down the cable. There is an optimal diameter that will balance the competing forces of input resistance and attenuation on the voltage response. For a constant current input, the optimal diameter should be chosen to make the electrotonic length of the cable $L \approx 3$ (Holmes, 1989). Because of reciprocity in a linear system, diameters that maximize the voltage at a distal tip for current input at the soma will also maximize the voltage at the soma for current input at the distal tip. If the input is a synaptic input, particularly a transient synaptic input, the optimal diameter occurs for much smaller values of L . However, reciprocity does not hold for synaptic inputs because of the nonlinear effect of the driving force, and this means that diameters cannot be optimal for signaling in both directions. This analysis in a passive system suggests that it is possible that the morphology of a cell may be optimized to maximize the effectiveness of synaptic inputs from a given location having a particular time course (Mouchet and Yelnik, 2004; Ermi et al., 2001). A similar analysis has been done recently regarding optimal diameters for the placement of gap junctions (Nadim and Golowasch, 2006).

Branching Effects on Current Flow and the “3/2 Rule”

While the 3/2 rule is useful for reducing complex trees to an equivalent cylinder, it is less clear what use such a branching pattern might have in nature. It is tempting to suggest that the 3/2 rule might provide optimal diameters to maximize the effectiveness of certain inputs, but currently there is no evidence for this. One might ask if the 3/2 rule is the most efficient means to distribute voltage across branch points or wonder what happens to voltage drops at branch points when diameters do not follow this rule. Suppose diameters satisfy a different power rule at branch points. Suppose the power p equals 1 (taper) or 2 (flare). What effect does this have?

If diameters satisfy the 3/2 rule, there is impedance matching at branch points and this will be lost if $p \neq 3/2$. The result is that voltage attenuation away from the soma will change abruptly at branch points. If $p = 1$, voltage attenuation will be less steep before the branch point than after it. If $p = 2$, voltage attenuation will be steeper before the branch point than after it. Such changes are difficult to see clearly in plots (figure 10.2b) without larger deviations from the 3/2 rule, but they could have implications for how strongly somatic inputs are felt in proximal regions compared with distal regions. Conversely, for an input at the tip of a distal branch, there will always be an abrupt change in the rate of steady-state voltage attenuation at the branch point (see figure 4 of Rall and Rinzel, 1973). Attenuation will always be more shallow proximal to the branch point as current flows from the distal branch into a thicker proximal process and a sibling branch, but this attenuation between branch points is steeper if $p = 1$ and more shallow if $p = 2$ compared with when $p = 3/2$.

The result of this is that the asymmetry in steady-state voltage attenuation from a distal dendrite toward the soma compared with attenuation in the reverse direction is increased when $p = 1$ and reduced when $p = 2$ compared with the case where $p = 3/2$. Changes in attenuation are also accompanied by changes in input resistance, with R_N increased when $p = 1$ and reduced when $p = 2$. Because of the competing roles of input resistance and attenuation for determining optimal diameters, as mentioned earlier, it is possible that differences in the value of p may reflect a difference in how inputs in different parts of the dendritic tree are weighted.

Dendritic Spines

Passive cable theory has several implications for the function of dendritic spines (see Rall, 1974, 1978 and the review by Shepherd, 1996) and we will only mention two. First, spines provide a particular case of asymmetric voltage attenuation described earlier. There may be considerable voltage attenuation from the spine head to the dendrite if the spine stem resistance is large. However, there is virtually no voltage attenuation in the reverse direction; any voltage change in the dendrite is felt in the

spine. This fact ruled out an early hypothesis for spine function: that spines exist to isolate inputs from each other electrically. Second, an input on a spine is less effective than an identical input on the dendrite. Although this is true for passive models, it may not be true if voltage-dependent conductances are activated in spines.

10.4 Estimating Electrotonic Parameters—Historical View

To make passive models of neurons, it is essential to have estimates for the values of the electrotonic parameters as well as some representation for cell morphology. The electrotonic parameter whose value is known with the most certainty is C_m . Cole (1968) called C_m a biological constant with a value of approximately $1.0 \mu\text{F}/\text{cm}^2$ and although this value is often used, it may be an overestimate, as discussed earlier. Axial resistivity, R_a , has been difficult to measure, but at least we believe we have good bounds for this parameter. However, values for R_m and a suitable morphology are more difficult to determine. Rall pioneered methods to estimate these parameters with the equivalent-cylinder model and we briefly review these.

Use of Time Constants to Estimate Electrotonic Parameter Values

Rall (1969) showed that the voltage response following the termination of a brief or sustained current input could be described mathematically as an infinite sum of exponential terms:

$$V = C_0 e^{-t/\tau_0} + C_1 e^{-t/\tau_1} + \dots \quad (10.5)$$

If values for τ_0 and τ_1 can be estimated from a voltage transient by curve fitting or (in those days) “exponential peeling,” then critical information about the cell can be obtained. First, τ_0 , being the longest time constant, is equal to the membrane time constant τ_m . Because $\tau_m = R_m C_m$, we can estimate R_m assuming that C_m is $1.0 \mu\text{F}/\text{cm}^2$. Second, Rall showed that if the cell can be approximated as an equivalent cylinder, then the electrotonic length of the cell can be estimated from the equation, $L = \pi/(\tau_0/\tau_1 - 1)^{1/2}$.

Although Rall derived several other formulas for L that involved use of the coefficients C_i and voltage-clamp time constants, this particular formula was the most widely used because the current clamp time constants were the easiest to obtain with confidence. With estimates of R_m and L , one could construct a simplified model of a cell in which it was not essential to have precise lengths and diameters of neuron processes as long as these values met the constraint of the estimated L value. Application of the Rall formula has provided substantial insight into the electrotonic structure of numerous types of neurons in scores, if not hundreds, of studies through the years (e.g., Nelson and Lux, 1970; Lux et al., 1970; Burke and ten Bruggencate,

1971; T. H. Brown et al., 1981a,b; Johnston, 1981; Stafstrom et al., 1984; Fleshman et al., 1988; Pongracz et al., 1991; Rapp et al., 1994; London et al., 1999).

What Do the Time Constants Represent?

In a cylinder, the time constant τ_1 represents an equalization of charge between the soma end of the cylinder and the distal end. Subsequent time constants represent additional harmonics of charge equalization between the two ends of the cylinder. It is this charge equalization that makes voltage decay (and voltage charging) initially faster in a cylinder than in an isopotential sphere, as mentioned briefly earlier. The τ_1 will be half the size of τ_0 for $L \sim 3$, one-tenth of τ_0 for $L \sim 1$, and one-hundredth of τ_0 for $L \sim 0.3$, so charge equalization will modify the voltage transient in all but the most electrotonically compact cells.

The advent of neuron reconstruction methods (chapter 8) has allowed modelers to use actual cell morphology in models, but for morphologies different from those of an equivalent cylinder, the time constants have a different meaning. In a compartmental model with N compartments, N time constants can be computed from the eigenvalues of the compartmental model matrix (the matrix \mathbf{A} in the representation $dV/dt = \mathbf{AV} + \mathbf{B}$). It turns out that τ_1 represents charge equalization between the longest tip-to-tip path in the neuron, τ_2 represents charge equalization between the second longest tip-to-tip path, etc. These time constants are associated with odd eigenfunctions (having odd or $+/ -$ symmetry with respect to the soma). Eventually there is a time constant, τ_{even} , associated with an even eigenfunction (even symmetry) with respect to the soma that represents charge equalization between a proximal membrane and a distal membrane. In a morphology represented by an equivalent cylinder, the coefficients of the time constants associated with odd eigenfunctions (τ_1 , τ_2 , etc.) are all equal to zero and the first time constant that appears with a non-zero coefficient after τ_0 is τ_{even} . However, if the morphology deviates from an equivalent cylinder, then these other time constants will not have zero coefficients. In neurons having terminations at different electrotonic lengths, such as pyramidal cells (apical versus basilar dendrites), some of these coefficients can be significant. The assumption made when applying the Rall formula is that the fitted τ_1 (sometimes called $\tau_{1\text{peel}}$) will equal τ_{even} , but this may not be the case if other time constants have significantly nonzero coefficients because the fitted τ_1 will most likely reflect the average of dozens of closely spaced time constants (W. R. Holmes et al., 1992).

Constructing Models from Morphology, τ_0 and R_N

Some labs began to apply the Rall methods after obtaining morphological and electrophysiological data from the same cell (e.g., Clements and Redman, 1989; Fleshman et al., 1988; Nitzan et al., 1990). Standard estimates of C_m and R_a , an estimate of R_m from the experimentally determined τ_0 , plus the reconstructed morphology

should in theory provide all that is needed to construct passive models of these cells. However, when this was done in practice, some discrepancies were quickly noted. In particular, the input resistance, R_N , measured experimentally did not agree with the input resistance value calculated with the model.

Three hypotheses were proposed to explain these discrepancies. First, perhaps the standard values assumed for C_m and R_a were not correct. However, the changes required were too large for this possibility to be readily accepted. Second, perhaps the electrotonic parameters, in particular R_m , were not uniform in the cell. Unfortunately not enough information was available to distinguish among various R_m distributions. Third, it is well known that membrane does not necessarily provide a tight seal around intracellular electrodes, and perhaps an electrode shunt could cause an artificially low τ_0 . The data seemed to be most consistent with this third possibility. This required estimation of an additional parameter, the soma shunt (Rose and Dagum, 1988; Rose and Vanner, 1988; Clements and Redman, 1989; Pongracz et al., 1991). Given that sharp electrodes introduce an artificial shunt at the soma, it became clear that earlier estimates of L obtained with the Rall formula overestimated the actual L by a factor of two in electrotonically compact cells and by 0–20% in electrotonically long cells (Holmes and Rall, 1992a). Furthermore, estimation of C_m , R_a , R_m , and the soma shunt cannot be done uniquely without accurate morphological and electrophysiological information from the same cell (W. R. Holmes and Rall, 1992b; Major et al., 1993a) and even then, it is possible to obtain several sets of parameter values that appear to fit the data equally well (Rapp et al., 1994).

Whole-Cell Patch Recordings

The development of the whole-cell patch-clamp technique allows one to avoid introducing an artificial shunt into the cell, but this recording method is not without its own issues. The technique may dialyze the cell, so it is important to have the appropriate medium composition in the electrode. In some of the early whole-cell patch data, values of input resistance and associated estimates of R_m and R_a were 5–10 times larger than estimates obtained with sharp electrodes. The reasons for this were that many of these recordings were done from immature neurons from young animals and the technique allowed more recordings from smaller cells to be successful; however, the biggest problem appears to have been cell dialysis. The use of the perforated-patch technique, along with better knowledge of what needs to go into the patch electrode (Kay, 1992), has brought estimates of these parameters back down substantially, although values remain higher than those obtained with intracellular electrodes (Pongracz et al., 1991; Spruston and Johnston, 1992; Staley et al., 1992). The use of whole-cell patch recordings has brought to light a fourth explanation of the discrepancies noted earlier that was known but largely ignored at the time—the accuracy of the morphological reconstructions. This is discussed further later.

10.5 Estimating Electrotonic Parameters—Recent Approaches

Recent efforts to estimate passive electrotonic parameters use whole-cell patch or perforated-patch recordings combined with morphological reconstructions from the same cell. A sample of these efforts for seven different cell types is discussed here. Patch recordings eliminate the soma shunt problem and this, together with complete morphological information from the same cell, removes a major source of non-uniqueness of parameter value estimates. After the experience with soma shunt caused by sharp electrodes, particular concern has been devoted to minimizing technical problems during experiments, such as those caused by electrode resistance and capacitance, and these are discussed at length in Major et al. (1994) and Roth and Häusser (2001).

The types of recordings that have been used in recent analyses are the response at the soma to a brief pulse of current (Thurbon et al., 1998; Roth and Häusser, 2001; Major et al., 1994; Trevelyan and Jack, 2002), short and long hyperpolarizing current pulses (Stuart and Spruston, 1998; Golding et al., 2005), or just long hyperpolarizing pulses (Chitwood et al., 1999). Brief pulses are presumed to be too brief to activate voltage-dependent conductances, leaving only a passive response for analysis. Responses to long hyperpolarizing pulses are typically done with H-channel blockers to linearize responses as necessary.

The morphological reconstructions were used to construct models and then parameter values for R_m , R_a , and C_m were fit in the models to match the voltage responses. Roth and Häusser (2001), Stuart and Spruston (1998), Golding et al. (2005) and Chitwood et al. (1998) used the multiple-run fitter in NEURON to do their fits, while other groups used direct fits with standard optimization methods. Three groups (Stuart and Spruston, 1998; Roth and Häusser, 2001; Golding et al., 2005) were able to record simultaneously from dendrites and the soma. Having recordings from multiple locations in the cell is exceptionally useful for fitting R_a and for determining whether R_m is uniform. Stuart and Spruston (1998) and Golding et al. (2005) could not find good fits to both dendritic and somatic recordings when R_m was assumed to be uniform; the fitted model responses decayed too slowly in the dendrite and too quickly at the soma. Much better fits were obtained by assuming R_m was lower in distal than in proximal regions.

Estimated parameter values from seven studies are given in table 10.1. The parameter values show some variability, but on average the C_m and R_a values are in line with those mentioned at the beginning of this chapter that were directly measured or calculated. The heterogeneity may reflect actual differences among cell types, differences in recording methods, problems with morphological reconstructions, or (except for the spinal cord study) uncertainty about appropriate compensation for dendritic spines.

Table 10.1

Passive electrotonic parameter value estimates from recent studies of seven different cell types

Study	Cell type	C_m ($\mu\text{F}/\text{cm}^2$)	R_a ($\Omega \cdot \text{cm}$)	R_m ($\text{k}\Omega \cdot \text{cm}^2$)
Trevelyan and Jack (2002)	Layer 2/3 cortical pyramidal (37°)	0.78–0.94	140–170	Soma shunt
Thurbon et al. (1998)	Spinal cord cells	2.4 ± 0.5	87 ± 22	5.3 ± 0.9
Roth and Häusser (2001)	Cerebellar Purkinje cells	0.77 ± 0.17	115 ± 20	122 ± 18
Stuart and Spruston (1998)	Layer 5 cortical pyramidal cells	1.1–1.5	70–100	$35 \rightarrow 5$ nonuniform
Major et al. (1994)	CA3 pyramidal cells	0.7–0.8	170–340	120–200
Golding et al. (2005)	CA1 pyramidal cells	1–2	139–218	$87 \rightarrow 20$ nonuniform
Chitwood et al. (1999)	Hippocampal interneurons	0.9 ± 0.3	Not available	61.9 ± 34.2

Parameter value ranges or means \pm SD are given.

10.6 Considerations for Constructing Passive Cable Models

When one develops a passive cable model to study a specific phenomenon, there are several steps one should take and there are many places where one can go wrong. In this and the following section we discuss some of the issues, make some recommendations, and discuss some potential pitfalls. Much of what we will say is also relevant for models that include active membrane conductances that are discussed in the next chapter.

Choosing a Morphology

One of the first issues to be resolved is what to use to represent the morphology of a cell. Earlier we discussed the equivalent-cylinder model, but some of its key assumptions—that dendrites all end at the same electrotonic distance and that diameters at branch points satisfy the 3/2 rule—are often violated. A reduced model with a simplified morphology that better captures the essential morphological characteristics of a cell could be useful for network models where computation time is an issue. One such simplification is to reduce the dendritic tree into an unbranched “equivalent cable” with variable diameter (Clements and Redman, 1989; Fleshman et al., 1988). Dendrites from a morphological reconstruction are divided into small segments having an equal electrotonic length, ΔX (or equivalently, for fixed R_m and R_a , equal $\ell/d^{1/2}$). Then the diameter of the equivalent cable, d_{eq} , at electrotonic distance X_j ($X_j = \sum_{i=1}^j \Delta X$) is determined from the diameters of all dendrites present at that distance by $d_{\text{eq}}(X_j) = [\sum_i d_i(X_j)^{3/2}]^{2/3}$. This “equivalent cable” has approximately the same membrane area, input resistance, and time constant τ_0 as the fully branched structure, but the voltage transients are not identical (Rall et al., 1992).

The equivalent-cable model has been used to quickly estimate electrotonic parameter values from transients (Clements and Redman, 1989) and to collapse those parts of the dendritic tree not of interest for the question being modeled (Burke et al., 1994). It also provides a means to collapse subtrees in cells with more complicated structures for more efficient computation (Stratford et al., 1989).

With simplifications such as these, a cell may be represented with 10–40 compartments. However, if a simplified morphology is truly sufficient, then why has nature made dendritic trees so complicated? In recent years, a number of studies have shown that morphology plays an important role in the computations done by a cell (Hausser and Mel, 2003; Krichmar et al., 2002; Schaefer et al., 2003b; van Ooyen et al., 2002). With the advent of more powerful computers, computation time is not a major issue for models with detailed morphology, particularly passive models.

Detailed morphological reconstructions are becoming more and more available for a variety of cell types in public databases. The effort of anatomists to produce these data and to make them readily available is very much appreciated by the modeling community. Data that were maintained on web sites of individual labs are now being gathered more centrally. For example, <http://neuromorpho.org>, an archive maintained by G. A. Ascoli (2006), has collected reconstruction data for over a thousand cells from fifteen different laboratories. Much of this reconstruction data can be saved in NEURON and GENESIS formats. These morphological reconstructions provide an appropriate anatomical structure for the cell type without making any simplifying assumptions about morphology and should be used whenever possible.

Selecting Electrophysiological Data for Estimating Electrotonic Parameter Values

It is not enough to take a morphological reconstruction and use standard or typical values for the electrotonic parameters R_m , R_a , and C_m in a model. Values for these parameters must be fit to electrophysiological data so that the model not only has a representative morphological structure, but also produces representative electrophysiological responses. Ideally, one would like to have morphological and electrophysiological data from the same cell (and this is essential if the goal is to estimate actual values for electrotonic parameters, as discussed in the previous section), but these data are available in only a few labs, and morphology databases typically have no recordings from the reconstructed cells. There is a great need in the modeling community for a public database containing actual electrophysiological recordings with a number of repeated trials, even if the detailed morphology of such cells is not known.

Assuming electrophysiological data can be obtained, what type of data would be most useful for parameter fitting? Because we are discussing passive models, we first of all want our data to be free of the influence of voltage-dependent conductances. Brief current pulses are often used because they should not activate voltage-

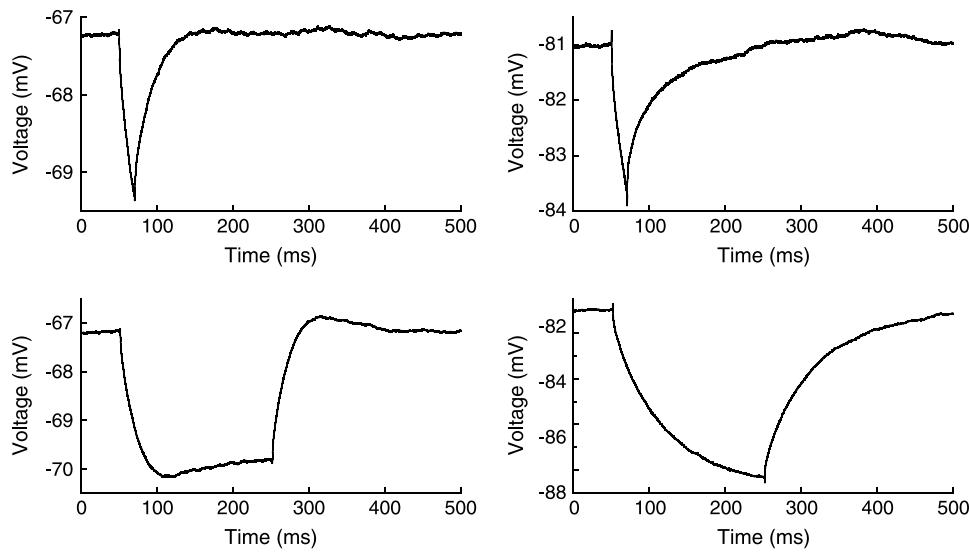


Figure 10.3

Experimental recordings in response to hyperpolarizing current pulses. Response to short (upper traces) and long (lower traces) hyperpolarizing current pulses from hippocampal CA1 pyramidal cells. (a) Cell in normal artificial cerebrospinal fluid (ACSF). (b) Cell with the H-channel blocker ZD-7288. The cell in ACSF shows the characteristic voltage sag in response to a long-duration current injection. The sag is eliminated in the other cell where the H-current is blocked.

dependent conductances. Weak hyperpolarizing and depolarizing current pulses can be used, and if we are fortunate, voltage changes with these currents will scale linearly with current amplitude. If not, then some cocktail of blockers of voltage-dependent conductances will have to be used. The voltage responses to short and long -50-pA current pulses injected into a CA1 hippocampal pyramidal cell are shown in figure 10.3a. The response shows the characteristic voltage sag seen in these cells that results from activation of the hyperpolarization-activated current, I_h . This current is blocked in another CA1 pyramidal cell by bath application of the H-channel blocker ZD-7288, as shown in figure 10.3b. The sag is gone and the voltage response is purely passive.

Second, we want to have voltage traces that provide sufficient information to estimate the electrotonic parameters. A long hyperpolarizing pulse will provide information about R_m and R_a , as well as input resistance, and the time course will provide information about R_m and C_m . A short hyperpolarizing pulse will provide information primarily about R_a and C_m , so both short and long current pulses should be used. In a passive system, it is not very useful to have data with current pulses of dif-

ferent amplitudes and the same duration except to check that the system is indeed linear and to catch possible problems caused by experimental noise, because no new information will be obtained. Recordings at sites different from the soma can provide useful information, particularly if the sites of the inputs are known and the morphology and electrophysiology come from the same cell, as noted earlier (Stuart and Spruston, 1998; Golding et al., 2005; Roth and Häusser, 2001). Other types of data, such as more complicated waveforms, direct current field stimulation (Svirskis et al., 2001), or impedance functions (Maltenfort and Hamm, 2004) may also be useful.

Compensation for Dendritic Spines

Many neuron types receive synaptic inputs on dendritic spines. A neuron can have many thousands of dendritic spines whose combined membrane area can be half the total membrane area of the cell. It is tedious to model every single spine explicitly and more tedious and generally not practical to measure spine dimensions for every spine on a cell (but see White and Rock, 1980). However, if one merely assigns standard values for R_m , C_m , and R_a to a morphological model without including spines or otherwise compensating for them, then the model will not provide an appropriate electrotonic structure. Input resistance will be overestimated and the results will not be representative.

There are two methods used in models to compensate for dendritic spines. The first is to increase C_m and reduce R_m according to total membrane area with spines included compared with total membrane area without spines (W. R. Holmes, 1989). For example, if the inclusion of spines on a dendritic segment increases total membrane area by 33%, then these spines can be included implicitly by multiplying C_m by 1.33 and dividing R_m by 1.33. Because spine density may not be the same in different parts of the cell, it is necessary to determine spine area and change R_m and C_m separately for each dendrite. A second method is to keep R_m and C_m the same on each dendrite, but increase the length and diameter of the dendrite to account for the extra membrane contributed by spines, but in such a way as to keep intracellular resistance (equivalently, length divided by diameter squared) the same (Stratford et al., 1989). In the earlier example where spines increase total membrane area by 33%, we let $F = 1.33$ and then multiply length by $F^{2/3}$ and diameter by $F^{1/3}$. It is quite simple to implement either of these two methods in simulation scripts.

These two methods for including spines work identically in passive models, but have very different implications for models with voltage-dependent conductances in dendrites. The first method, where R_m and C_m are changed, compensates for spines with passive membranes, whereas the second method, where length and diameter are changed, assumes that spines have the same densities of voltage-dependent conductances as the dendrite. The process of collapsing spines that are not passive and do not have the same voltage-dependent properties as the dendrite to which they

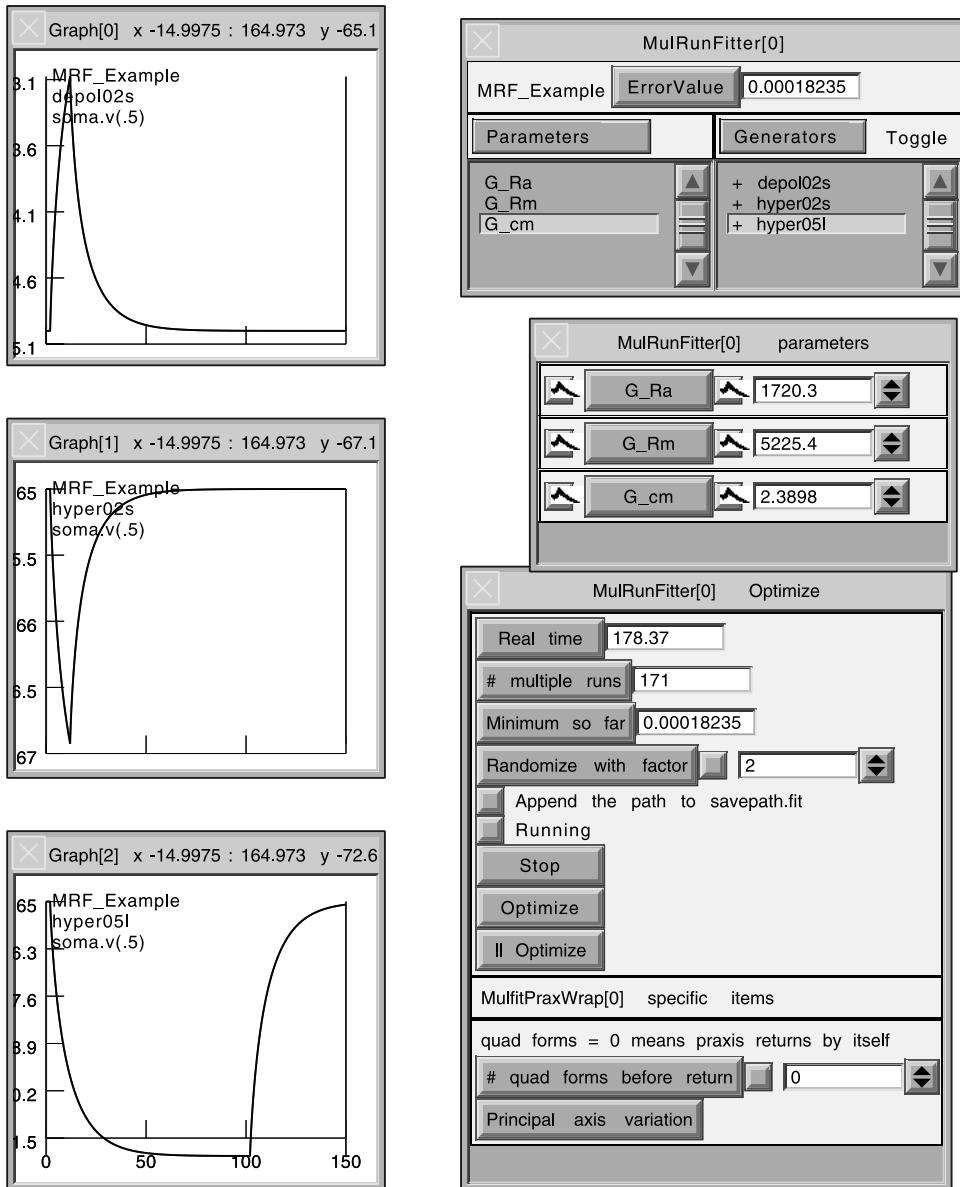


Figure 10.4

Local minima and sensitivity issues during the parameter search. Example using NEURON's multiple-run fitter. The “neuron” has a simple Y-branched apical dendrite, a basilar dendrite, and an axon (all passive). Spines were included on apical dendrites. The “experimental data” were generated with $R_m = 12,000 \Omega\text{cm}^2$, $R_a = 160 \Omega\text{cm}$, and $C_m = 1.0 \mu\text{F}/\text{cm}^2$ for all compartments. The fitter converged to these values with error 9e-10. However, with different starting values, there was convergence to a local minimum with parameter values that gave a very good-looking fit (fit and target traces overlap; parameter values and error shown in the figure). R_m , R_a , C_m combinations of (15,629, 53,296, 0.65739) or (14,811, 72,23, 0.72814) have low error and provide good fits visually, but were not considered final solutions by the fitter. The files to run this example are available on the book website at <http://www.compneuro.org>.

are attached is relatively straightforward, but requires that one be careful in the implementation.

Estimation of Electrotonic Parameter Values

After choosing a morphology for the model, selecting electrophysiological data and arranging to compensate for spines, if necessary, the next step is to fit electrotonic parameter values to make the model responses match the data. There are several ways this can be done. One method that we have found particularly useful, and which has been used by several groups, as noted earlier, is the multiple-run fitter available in the NEURON simulator (figure 10.4). NEURON uses the principal axis method (PRAXIS), a variant of the conjugate gradient method developed by Brent (1976), by default; see Shen et al. (1999). Chapter 2 discusses other parameter search methods that can be added to the NEURON multiple-run fitter or else used independently.

10.7 Problems, Pitfalls, and Recommendations

Problems with Morphological Reconstruction Data

The use of morphological reconstructions for modeling studies is recommended because a fixed and accurate morphology greatly reduces the number of degrees of freedom for the model. However, despite the enormous effort that is put into doing a morphological reconstruction and the improvements in reconstruction methodology through the years, the fact remains that precise morphological measurements are difficult to obtain (see chapter 8). Process diameters in particular are very difficult to measure accurately because many diameters are near the limit of resolution of the light microscope. Diameter measurements given in databases rarely have more than one digit of precision. Other potential problems are tissue shrinkage, artificial compression of tissue, and the difficulty of reconstructing lengths accurately in the z -axis direction as processes are followed from one plane of focus to another. Recent studies have analyzed reconstructions of the same cell type compiled by different labs (Ambros-Ingerson and Holmes, 2005; Scorcioni et al., 2004; Szilagy and De Schutter, 2004) and have found that morphological parameters such as total dendritic length, membrane area, and volume were similar among cells reconstructed in the same lab but were very different among different labs. This variability in measurements among labs is unlikely to be explained by differences in animal strain or age.

The variability in reconstructed measurements has consequences for estimates of electrotonic parameters. Recall that R_m is the membrane resistivity of a unit area of membrane in units of ohms times square centimeter, C_m is membrane capacitance per unit of area in units of microfaradays per square centimeter, and R_a is axial

resistivity through a unit cross-sectional area per unit of length in units of ohms times centimeter. Consequently, if the reconstruction diameters are uniformly off by a factor of x , then the fitting procedure will return estimates of R_m , C_m , and R_a equal to the actual R_m multiplied by x , the actual C_m divided by x , and the actual R_a multiplied by x^2 . If reconstruction lengths are uniformly off by a factor y , then the fitting procedure will return estimates of R_m , C_m , and R_a equal to the actual R_m multiplied by y , the actual C_m divided by y , and the actual R_a divided by y . R_a estimates are particularly vulnerable to error because of the x^2 dependence of its estimate on diameter and the fact that diameter is difficult to measure precisely. Despite the consequences of reconstruction errors for the electrotonic parameter estimates, reconstructed morphologies should still be used in the models. They provide a characteristic anatomical structure of the cell type being studied, and issues with reconstructions can be overcome with parameter fitting, as discussed next.

It Is Essential to Fit Parameter Values with Data

If one blindly uses standard values for R_m , C_m , and R_a in models based on reconstruction data, the results may not be representative for the cell type being studied. Even if the morphological data and the electrophysiological data come from the same cell and the morphological reconstruction is perfect, parameter estimation is still necessary for two reasons. First, although we know ranges of values for the electrotonic parameters, the “standard” values are not known precisely enough that one can confidently say that a model with such values will produce representative results. Second, even if the dendrites are reconstructed perfectly, it is unlikely that the number of dendritic spines or the total spine area is known with much precision. Parameter fitting will still be needed to account for dendritic spines.

Some models may be based on morphological data obtained from a database and electrophysiological data obtained from a different cell. The assumption is that the morphology chosen is typical and provides an appropriate structure for the cell and that the electrophysiological data are also typical. Here parameter fitting is necessary to compensate for (1) reconstruction issues, (2) an unknown number of spines, and (3) the fact that the two types of data come from different cells. Parameter fitting can make the morphological cell model electrotonically equivalent to the experimental cell (W. R. Holmes et al., 2006). Clearly, there is heterogeneity among cell types, and the morphology and cell responses of the reconstructed cell and the experimental cell may in fact be different. However, the morphological reconstruction provides a basic anatomical structure that is common for the cell type, and with fitted electrotonic parameter values, a model based on this reconstruction will provide representative responses for that cell type. This is really what we want even though the actual electrotonic parameter values in the model may not be typical or standard values.

Local Minima, Uniqueness, and Sensitivity Problems

Issues associated with parameter fitting are discussed in detail in chapter 2. It is not clear how often the problem of local minima appears when fitting passive models, but in our experience this has not been much of an issue when there are suitable bounds on the parameter space. In the very few cases where we have seen convergence to multiple solutions, the fitting error has clearly indicated the better solution. However, finding local minima in the simple example in figure 10.4 proved to be much easier than with real data. For example, with starting values of 100, 1,200, and 1 for R_a , R_m , and C_m , the method converges to $R_a = 1720.3$, $R_m = 5225.4$, and $C_m = 2.3898$. The model with these parameter values appears to overlap the “experimental” traces visually, as shown in figure 10.4. The error is very small, although it is orders of magnitude larger than the error with the “true” values. In situations like this, it is helpful to set physiological bounds on parameter values, but even then the method may sometimes get stuck on one of the bounds. It is advisable to try several different starting value combinations. The issue of nonuniqueness of parameter fits is discussed in chapter 12, section 3. For passive models, the fitting procedure described typically assumes that R_m , R_a , and C_m are uniform, but this does not have to be the case. It is likely that equally good (or better—Stuart and Spruston, 1998; Golding et al., 2005) solutions can be obtained if this assumption is relaxed and some functional form is used to describe how the values of these parameters change with distance from the soma. If these parameters are uniform, morphology provides a significant constraint limiting nonuniqueness. For example, in the equivalent-cylinder model values of L , R_N , and τ do not fix R_m , R_a , and C_m uniquely unless length and diameter are also specified. For more complex models, morphology plus experimental voltage traces appear to fix uniform R_m , R_a , and C_m values uniquely, but this has not been proven.

A larger problem is sensitivity. When the example in figure 10.4 is run, solutions that do not look bad visually appear long before convergence. If the fitter is stopped before convergence at one of these not-so-bad solutions, the parameter values may be significantly different from the “true” or final values. Experimental data contain noise, and there is no guarantee that noise will not cause very different parameter values to appear as the final solution (Major et al., 1994). If noise is truly random, then use of multiple sets of experimental data with different protocols might minimize this problem.

Recommendations for Modeling

Ideally one should try to get a morphological reconstruction and electrophysiological data from the same cell. Failing that, morphology can be obtained from a public database and electrophysiological data for the same cell type can be obtained

separately, but these two types of data should be matched for the same strain of animal at the same age. Passive electrotonic parameters in the model should be fit to match multiple voltage traces from the experimental cell, to make the modeled cell electrotonically similar to the experimental cell. Blind use of standard values for electrotonic parameters is not likely to yield representative results. Because of heterogeneity within a cell type in both morphological reconstructions and electrophysiological data, modeling studies should use multiple morphologies and multiple sets of experimental data; results generated from just one cell morphology and one set of experimental data should not be considered robust.

Acknowledgments

I would like to thank Jose Ambros-Ingerson for comments and suggestions on this chapter and Larry Grover for the experimental data shown in figure 10.3.

11 Modeling Complex Neurons

Erik De Schutter and Werner Van Geit

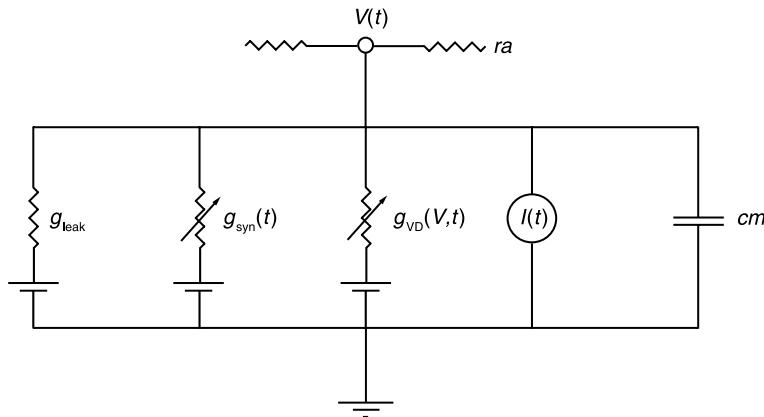
This chapter combines most of the modeling tools described in the preceding chapters to build models of complete neurons with active membranes. It builds on chapter 10, where one-dimensional passive cable modeling was introduced. The difference with the models presented in chapters 5 and 7 is that here the models have multiple compartments. These are embedded with voltage-gated channels that are simulated with Hodgkin-Huxley-like or more complex equations (chapter 5). The models can be simple, containing just a few compartments (section 11.2), or may represent the complete morphology (chapters 8–9) of the neuron (section 11.3). Often it is of interest to simulate synaptic input (chapter 6) to the isolated model (section 11.4), or the model can be embedded in small or large network simulations (chapters 12–13). We will refer extensively to the two most commonly used compartmental simulator software packages, GENESIS and NEURON (see the software appendix).

11.1 Compartmental Modeling

Almost all complex neuron models are simulated with compartmental modeling methods. This method was introduced by Rall (1964) and corresponds to the discretized version of the PDE describing a one-dimensional cable model (equation 10.2). The dendritic structure is subdivided into many compartments so that its morphology is faithfully replicated and so that each compartment is small enough to be considered isopotential (see later, this section). The change of voltage in each compartment (figure 11.1) is described by an ODE that is similar to equation (5.1) but with an extra term describing the current flow between connected compartments:

$$cm_i \frac{dV_i(t)}{dt} = - \sum_{k=1}^n g_{i,k}(V, t)[V_i(t) - E_k] + \sum_{j=1}^m \frac{V_j(t) - V_i(t)}{ra_{i,j}} + I_i(t). \quad (11.1)$$

Note the addition of the i subscript to all variables and parameters, denoting that this equation refers to one compartment of many, with parameters specific to each (e.g., cm_i).

**Figure 11.1**

Circuit diagram of an isopotential compartment. The compartment is coupled to other compartments over an axial resistance ra ; all other currents flow over the cell membrane (represented by the box). The membrane current is given by the sum of (from left to right) a leak current, synaptic currents, voltage-dependent currents, injected current, and a capacitative current (over the membrane capacitance cm). The first three currents are determined by a conductance (g) coupled to a battery (corresponding to the reversal potential E).

The first term on the right-hand side describes the membrane currents. These are assumed to operate in parallel (figure 11.1), independent of each other, and can therefore be summed. They comprise several types of currents, which in equation (11.1) are given as n different ohmic currents k , each with a conductance $g_{i,k}$ and a reversal potential E_k . In this formulation the passive leak resistance (r_m in chapter 10) has to be specified as a separate conductance ($1/r_m$) with a leak reversal potential. This is the way NEURON implements compartments; in GENESIS the leak current is part of the compartment variables (Rm and Em fields). The other currents will have a variable conductance g , which in the case of a Hodgkin-Huxley-like equation, can be described by the following generic equation:

$$g_{i,k}(t) = \bar{g}_{i,k} m^x h^y, \quad (11.2)$$

which is a derivation of equation (5.4) and depends on additional ODEs for the activation (m) and (h) gating variables (equations 5.6 or 5.8). Sometimes a third calcium-dependent activation gating variable z is multiplied with the right-hand side of equation (11.2) (De Schutter and Bower, 1994a; Khaliq et al., 2003). This has the conceptual advantage of having calcium activation as a separate variable. However, this simplification will scale the conductance by the calcium concentration while, instead, in many channels it causes a rightward shift of the activation curve (e.g., the BK Ca^{2+} -activated K^+ channel; Moczydłowski and Latorre 1983; X. P. Sun et al., 2004).

In many cases, in particular for calcium channels, the ohmic assumption does not hold and the GHK equation (equations 5.12–5.13) should be used. In that case the corresponding ohmic term in the summation over all membrane currents of equation (11.1) is replaced by

$$\overline{P_{i,k}}m^xh^yG(V_i, C_{o,i}, C_{i,i}), \quad (11.3)$$

with $G(V_i, C_{o,i}, C_{i,i})$ defined as in equation (5.13). Finally, this summation also includes all synaptic currents, described again as either ohmic (equation 6.10) or its GHK equivalent (equation 4.2).

The second term of equation (11.1) describes the axial current flow between compartment i and the m compartments j it is connected to. The size of m depends on the topology of the model: a compartment at the end of a branch will have a single neighbor, one in an unbranched section will have two, and one at a branch point at least three. The final term is optional and is used to simulate a current injection, through an electrode, into the compartment.

Compartment Parameters

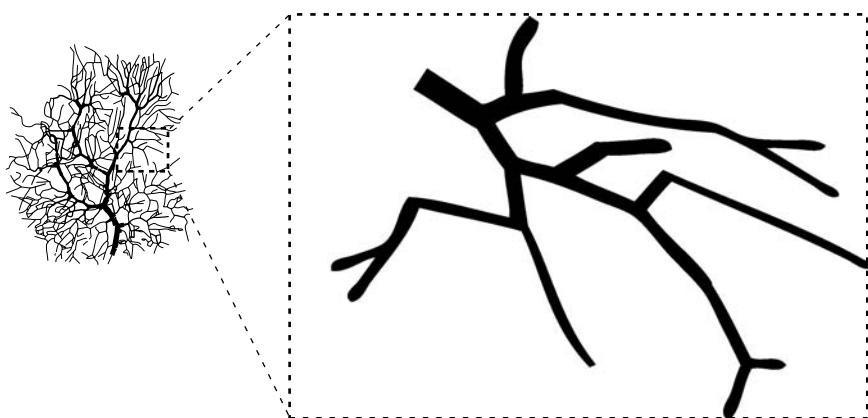
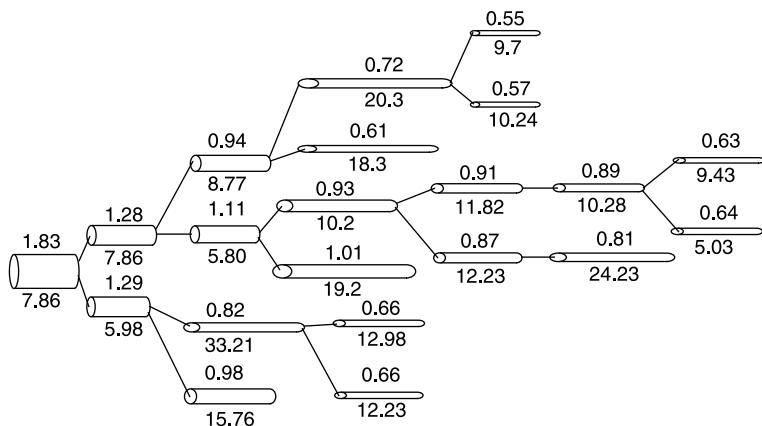
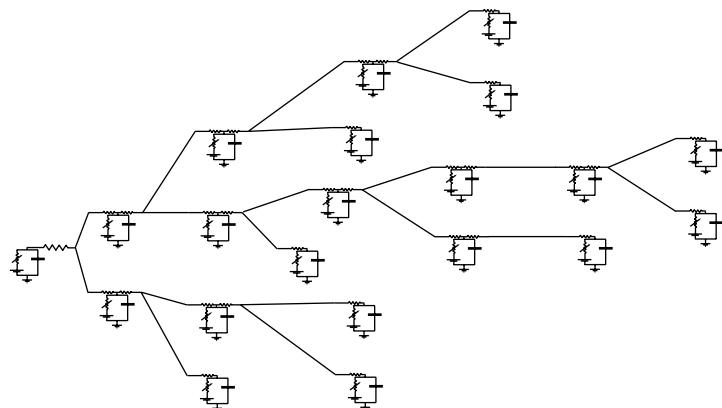
Equation (11.1) contains a number of parameters that will be specific to each compartment. We briefly develop the additional equations needed to compute these parameters, but in practice these computations are handled by the simulation software that combines user-defined values for the cable parameters with the neuron morphology read from a file (chapter 8, section 4). As described in chapter 10, section 1, the passive properties of a neuron can be described by three cable parameters: C_m , R_m , and R_a , which are defined for units of surface area or length. Assuming that the compartment is cylindrical with length l_i and diameter d_i , this gives the following equations:

$$\text{surface: } s_i = \pi d_i l_i \quad (11.4)$$

$$\text{membrane capacitance: } cm_i = C_m s_i \quad (11.5)$$

$$\text{cytoplasmic resistance: } ra_{i,j} = \frac{4R_a l_i}{\pi d_i^2}. \quad (11.6)$$

The computation of cytoplasmic resistance depends on how one maps compartments onto the basic electrical circuit. There are two ways in which this can be done: using asymmetric compartments (GENESIS nomenclature, called three-element segments in Segev et al., 1985) or symmetric compartments (four-element segments; this is the only type available in the NEURON simulator). The difference between these two approaches is where V_i is computed; at the boundary of the compartment (asymmetric) or at its center (symmetric), the latter approximation is more

**a****b****c**

accurate. Since V_i is assumed to be constant across the compartmental membrane, this may seem rather irrelevant, but it does determine how the cytoplasmic resistance should be computed. For the asymmetric case, this resistance is taken across one resistor with diameter d_i and length l_i (equation 11.6). For unbranched symmetric compartments, it is taken across two resistors in series: the sum of two equations (11.6) with, respectively, diameters d_i and length $l_i/2$ and d_j , $l_j/2$. For branch points on a symmetric compartment, the computation of ra becomes more complicated (Segev et al., 1985; see equation 1.56 for terminals).

In addition, the maximal conductances $G_{i,k}$ of all membrane currents are usually also specified for units of surface and need to be scaled for each compartment:

$$\text{membrane conductances: } \bar{g}_{i,k} = G_{i,k} s_i = \rho_{i,k} \gamma_k s_i. \quad (11.7)$$

The second equation of 11.7 describes the actual underlying biophysical system; each conductance is assumed to be caused by many single channels with a density $\rho_{i,k}$ and a single channel conductance γ_k . Though seldom done, it makes comparisons with experimental data easier if the parameter controlling maximum conductance is expressed with the channel density ρ instead of using G .

Accuracy of Model and Simulation

In a detailed compartmental model the three-dimensional morphology is mapped onto a one-dimensional branched cable with the same topology and sizes as the original model, and this cable is then discretized into compartments (figure 11.2). This discretization requires that single compartments be considered isopotential, which only holds when they are small enough. Compartment length (l_i) is determined by two factors: the morphology and a maximum electrical size. One tries to represent all the branch points in a morphology accurately, but it may not be necessary to keep all the small changes in diameter, taking into account the inaccuracies in the measurements (chapter 8). In combining multiple measurements along an unbranched segment into a single compartment, care should be taken to give the long compartment a size that results in the same total membrane surface (equation 11.4) and ra (equation 11.6) as the sum of the corresponding values for many shorter compartments. For the maximum electrical size, a rule of thumb is to take a size of 0.05λ for passive compartments, which allows for fairly long compartments (80 μm for a

Figure 11.2

Relation between neuronal morphology and a compartmental model. (a) Complete drawing of a reconstructed Purkinje cell neuron (De Schutter and Bower, 1994a); inset: enlarged drawing of the branch represented in panels b and c. (b) Cable representation of this branch as a system of cylindrical segments. For each segment, the diameter and length are indicated in micrometers above and below the cylinder, respectively. (c) Compartmental representation of this branch as set of coupled circuit diagrams similar to the one shown in figure 11.1. Note that because of the short length of the cylindrical segments in b, no further subdivisions were made. Figure inspired by a similar one in Segev and Burke (1998).

1- μ m-thick Purkinje cell dendrite; Roth and Haussner, 2001). In the case of active membrane, the changing level of activation of the voltage-gated channels, $\sum_{k=1}^n g_{i,k}(V, t)$ replaces r_m in equation (10.4). This makes λ variable (Holmes and Woody, 1989; Bernander et al., 1991; Rapp et al., 1992) and it is no longer possible to define an optimal value for the maximum electrical size. A possibility is to start to discretize at 0.02λ and check whether the model's behavior changes if smaller spatial discretization steps are used. It is easy to change spatial discretization in NEURON, but GENESIS is less flexible in this aspect.

Simulations are usually run with time steps on the order of 10 μ sec (but see chapter 4, section 2 if calcium diffusion processes are included). For both the spatial and time steps, the optimal values are those at which no changes in the model's behavior are observed when the step sizes are gradually made smaller. This should be checked repeatedly while model parameters are being changed. Note that one does not need to check the time step for adaptive time step methods because these methods take care of this automatically (Lytton and Hines, 2005; chapter 1, section 6).

Solution Methods

The voltage in each compartment is computed by equation (11.1) and for the complete model this leads to a system of coupled ODEs that are solved by numerical integration methods (chapter 1, section 6). In general, this is a stiff system for which implicit solution routines like the (fixed time step) Crank-Nicholson method are preferred. Ordering the compartments in a specific way introduced by M. Hines (1984), which is performed automatically by most simulation packages, reduces the number of off-diagonal elements and decreases the complexity of the solution matrix. An extensive analysis of the Crank-Nicholson solution method, its boundary conditions, and its accuracy can be found in Mascagni and Sherman (1998). Higher-order, adaptive time step methods are available in NEURON (Lytton and Hines, 2005); these will perform well for slowly spiking neurons.

To solve equation (11.1), the conductances $g_{i,k}$ and, if taken as a variable, the reversal potentials E_k must be computed also. Computing $g_{i,k}$ introduces a second set of ODEs: equations (5.6) or (5.8) in the case of Hodgkin-Huxley-like channels. This could make integration cumbersome because V_i appears in the equations describing the rate factors, requiring an iterative solution of the two sets of ODEs for each time step until V_i converges. However, because the time constants of ion channel gating are several orders of magnitude larger than the time step, one can assume for their calculation that V_i is constant over one integration step. This makes it possible to solve the two sets of ODEs in parallel (Mascagni and Sherman, 1998; Moore and Ramon, 1974), a solution method implemented in GENESIS and NEURON.

To speed up computation of the gating equations, the rate constants (equation 5.7) are often precomputed and stored in tables (De Schutter and Beeman, 1998). This

requires specific settings in GENESIS and NEURON. Such tabulation will be quite accurate for voltage-dependent equations if steps of 0.1 mV or smaller are used, but it is problematic for calcium-dependent gating mechanisms because small enough concentration steps require a huge table to cover the full range of possible intracellular calcium concentrations.

11.2 Phenomenological Models with a Reduced Morphology

Point neuron models (chapter 7) assume that all currents flow over a single patch of membrane. This seriously limits the number of phenomena these models can reproduce. For example, they cannot simulate dendritic spike initiation or synaptic integration because both phenomena are dependent on voltage gradients within the neuron. It may not be necessary, however, to simulate the complete morphology of a neuron. Using a reduced model will greatly reduce the number of model parameters needed and increase computation speed. However, one has to decide on how to reduce the morphology, which requires tradeoffs.

Limitations of Reduced Morphologies for Active Models

The equivalent-cable approach can be used to collapse a passive dendritic tree into an unbranched cable. This can be done approximately (e.g., Traub, 1979; Traub et al., 1991) or more accurately by varying the cable diameter (Fleshman et al., 1988; chapter 10, section 6), but this method does not generalize easily to active dendrites. The main problem is that the local input resistance in the “reduced” dendrite is different from that in the branches of the original one. The dendritic compartments of the reduced model will have quite large membrane surfaces and thus a low input resistance, whereas the small branches in the original dendrite have a very high input resistance. Consequently, when ionic channels are added to a reduced dendrite, their density must be higher to achieve the same voltage changes and evoke, for example, a dendritic calcium spike. Unfortunately this is not simply a scaling problem because the local input resistance has become voltage dependent. The only solution is to tune the channel densities specifically for this reduced model, possibly using automated search methods (chapter 2). This also applies to synaptic conductances (\bar{g}_{syn}) that, if the model is to be used in a network simulation, have additional restrictions because of network scaling requirements (chapter 13, section 3).

In general, parameter values for such models are difficult to relate to experimental data and the currents computed by the model are not realistic. Therefore, in the end it may be more sensible to give up the notion of morphological constraints on the model and consider phenomenological models with multiple compartments. Such models have the same disadvantages but are more flexible to use.

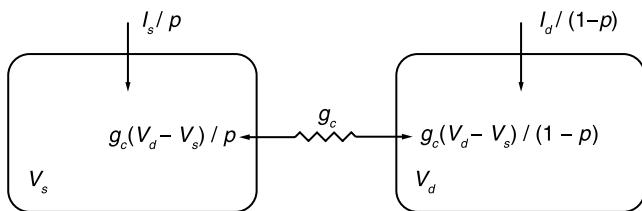


Figure 11.3

Schematic representation of the Pinsky and Rinzel (1994) model. The left box represents the somatic compartment, the right box the dendritic one. See text for an explanation of the symbols.

Phenomenological Models with Multiple Compartments

Here the definition of a compartment differs from that used in equation (11.1). Instead it follows the definition used in pharmacokinetics, systems theory, etc. (see also chapter 3, section 2), in being an abstract representation of a chemically or electrically homogeneous part of the model, without reference to a specific shape or size. Two parameters relate to morphology, but in a very abstract way: the relative size of the compartments and the coupling constant, which determines the interaction between the compartments.

While such an approach actually predates much of the “compartmental” modeling described in section 11.1 (Perkel et al., 1981), it was reintroduced to the modeling community by the work of Pinsky and Rinzel (1994), who could approximate the full firing behavior of an active-membrane nineteen-compartment cable model of a CA3 pyramidal neuron (Traub et al., 1991) with just two active compartments and a coupling conductance. This was achieved through separation of the voltage-gated channels over two compartments: a somatic one with currents supporting a classic action potential, and a dendritic region with calcium and calcium-activated currents able to generate a dendritic spike. The coupling between these two compartments allows the model to generate burst firing. A somatic action potential triggers the dendritic spike; current flowing back from the dendrite then triggers the second action potential, etc. A single compartment expressing all the voltage-gated channels cannot generate similar calcium spikes because a somatic afterhyperpolarization is required in order to remove the inactivation of the sodium channel at the same time that the dendritic spike is depolarizing the dendrite (Pinsky and Rinzel, 1994).

The behavior of the model (figure 11.3) is governed by two equations describing the change of somatic and dendritic voltage V_s and V_d :

$$cm_s \frac{dV_s(t)}{dt} = - \sum_{i=1}^n g_{i,s}(t)[V_s(t) - E_i] + \frac{g_c[V_d(t) - V_s(t)] + I_s(t)}{p} \quad (11.8)$$

$$cm_d \frac{dV_d(t)}{dt} = - \sum_{i=1}^k g_{i,d}(t)[V_d(t) - E_i] + \frac{g_c[V_s(t) - V_d(t)] - I_{\text{syn}}(t) + I_d(t)}{1-p}. \quad (11.9)$$

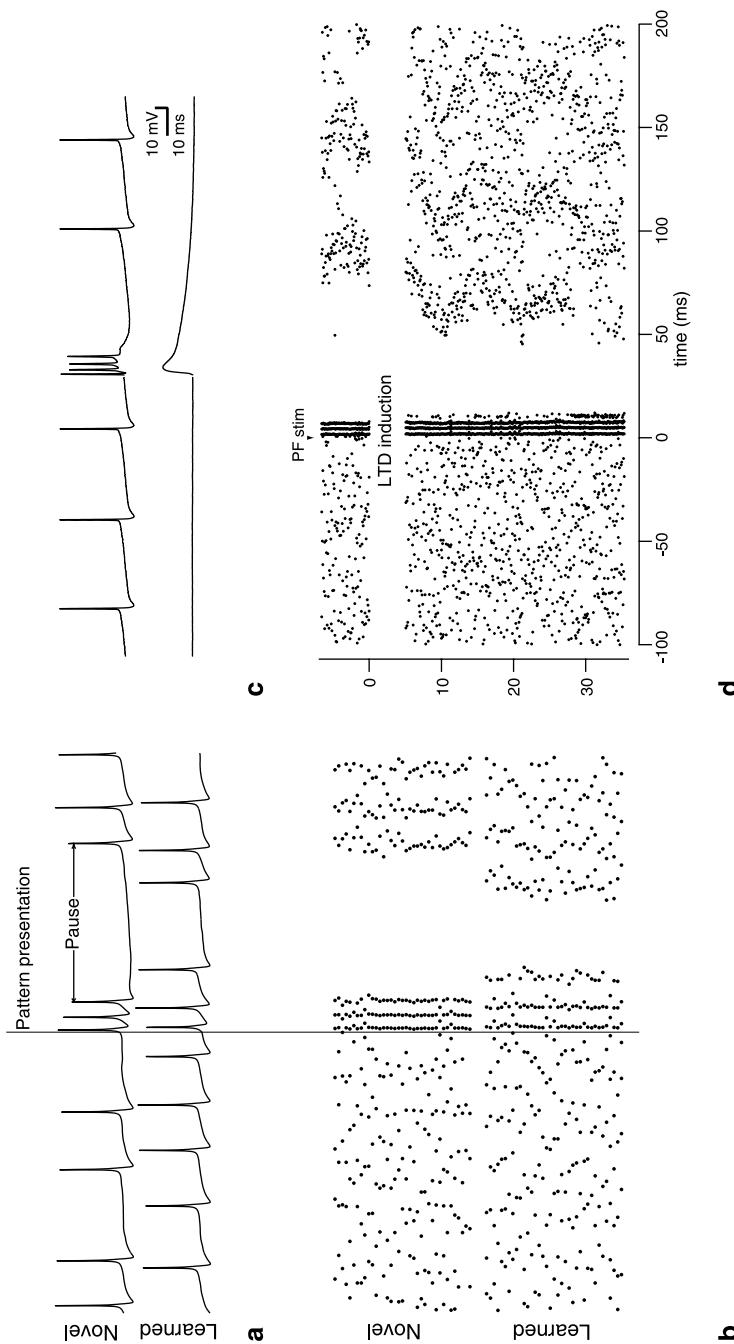
The two compartments are connected to each other by a coupling conductance g_c . All currents are scaled according to the relative contribution of somatic and dendritic compartments (p and $1 - p$; the scaling for the voltage-dependent currents is factored in g_i). The spatial components of the model are embedded in the parameters p and g_c , which have no direct relation to the actual size or shape of the neuron. In fact, a morphologically realistic value for g_c is too weak to produce bursting behavior. A typical value for p is 0.5, which indicates that the somatic compartment represents more than the morphological soma alone.

Besides its compactness, the model also has the advantage that its behavior can be related to simple parameters. Since g_c is not determined by morphology, it can be considered a free parameter and, depending on its strength, different bursting and firing patterns can be observed (Pinsky and Rinzel, 1994). Similar efforts to reduce complex existing models have shown that better results can be achieved with a reduced model with three to four compartments (Davison et al., 2000). Automated parameter search methods (chapter 2; Van Geit et al., 2007) should make this type of model reduction much easier. Another application of this approach is to de novo models, with the intention of building an approximate model that can be used to study the underlying dynamics of neurons where the interplay between soma and dendrite determines excitability (Fernandez et al., 2007).

11.3 Morphologically Detailed Compartmental Models

Although simplified models can relate neural behavior to simple parameters, these parameters are not easily compared with experimental measurements. Moreover, they do not allow study of the interaction between morphology and excitability in integration of synaptic input (Häusser and Mel, 2003). For such studies, or to simulate, for example, the biophysics of the interaction between a backpropagating action potential and synaptic activation in STDP (chapter 6, section 6) a morphologically detailed compartmental model of the neuron is required.

Such compartmental modeling may seem plug-and-play. Once the modeler has collected the morphology, cable parameters, channel kinetics, etc., they are plugged into the favorite simulator and the model is ready. In practice things are not so simple. The data may be difficult to find and, once obtained, often is not in a form suitable for modeling. They first need to be converted into equations and since these usually take standard forms (section 11.1), this means finding the correct parameters. Several of these issues have been covered in preceding chapters. Here we bring



all of them together and give practical advice, but we start with a short review of the literature.

Historical Overview and Current State

Collaborators of Rodolfo Llinás made the first large active neuron models. Andras Pellionisz developed a Purkinje cell model (Pellionisz and Llinás, 1977) and Roger Traub made a model of a hippocampal pyramidal neuron (Traub and Llinás 1977); in both models stylized dendrites were used. It wasn't until standardized programs like GENESIS and NEURON became available that these efforts were replicated, now using real morphologies based on microscopy reconstruction (chapter 8). We developed a reference model of the cerebellar Purkinje cell (De Schutter and Bower, 1994a,b) that was used to investigate the interaction between dendritic excitability and synaptic input, first of excitation (De Schutter and Bower, 1994c; De Schutter, 1998), then inhibition (Jaeger et al., 1997; Solinas et al., 2006) and recently even of synaptic plasticity (Steuber et al., 2007; figure 11.4). Around the same time, a reference model of a cortical pyramidal neuron was produced (Mainen et al., 1995).

While both of these models had a high impact on the community and continue to be used today, surprisingly few "new" models have been developed in the past decade. In fact, an amazing number of studies based on reconstructed neurons still combine active soma with passive dendrites (Roberts et al., 2006; Nenadic et al., 2003) or dendrites expressing just one channel type (Bekkers and Häusser, 2007; Kortaleski et al., 2006). Recent complete models, to some degree validated against experimental data, are available for CA1 pyramidal neurons (by Poirazi et al., 2003; a continuing series of models by Migliore and colleagues, Migliore et al., 1999, 2005a; Gasparini et al., 2004; Watanabe et al., 2002; and a recent model by Gold et al., 2006), of CA3 pyramidal neurons (Migliore et al., 1995), of globus pallidus neurons (Günay et al., 2008), of thalamic reticular neurons (Destexhe et al., 1996) and of deep cerebellar nuclei neurons (Steuber et al., 2009). In addition, the Blue Brain Project has been developing a large collection of morphologically detailed models of cortical neurons (Druckmann et al., 2007; Markram, 2006).

Figure 11.4

Experimental verification of Purkinje cell model predictions. Simulation (a, b) and experimental verification in slice (c, d) of the effect of learning through long-term depression (LTD) of parallel fiber (PF) synapses on the response of a Purkinje cell model (De Schutter and Bower 1994a,b) using the morphology in figure 11.2a. The main conclusion of the study (Steuber et al., 2007) was that the only significant change caused by LTD to the firing response is the shortening of the pause following a short burst of spikes. Panel a shows the spiking response before (novel) and after learning (learned) through LTD in the model. For comparison in c, an experimental spiking pattern similar to that after learning is shown with the corresponding EPSP below. Panels b and d show raster plots of the spiking response to multiple pattern presentations (model, b), or to multiple PF stimuli presentations before and after LTD induction (experiment, d). (Modified with permission from Steuber et al., 2007.)

One can wonder why the number of available models grows so slowly, because improved computer technology has made it so much easier to run them. The increased speed of today's computers makes the models not only run faster—simulations shown in the original Purkinje cell model paper (De Schutter and Bower, 1994a) that took an hour or more to run now finish in less than a minute and this could be reduced even further by parallelizing the model (Hines et al., 2008b)—but it has also enabled new technologies like automated parameter search methods (chapter 2). Despite these advances, the creation of a fully active compartmental model remains a lot of work, and automated parameter fitting is not a universal panacea. Moreover, the standards that a good model is expected to achieve have increased significantly and several of the models listed here are now considered flawed. These challenges may have discouraged many colleagues from embarking on the task of building detailed models. Nevertheless, we hope that by outlining the different steps that need to be taken to achieve this goal, this chapter will encourage a more widespread use of the method.

What Is a Good Model?

First of all, expectations should be realistic because otherwise the model creator may become disappointed. A model is always an approximation of reality and even if it is not perfect, it may still be a useful tool. For example, although we know that the original Purkinje cell model (De Schutter and Bower, 1994a) did not replicate important spiking mechanisms, such as the subsequently discovered resurgent sodium current (Khaliq et al., 2003), it still made a correct prediction of the experimentally observed change in spiking response to large parallel fiber patterns after the induction of long-term depression of synaptic input (Steuber et al., 2007; figure 11.4). Neither should one expect a model to reproduce every single experimental observation published. This is unrealistic because some results may not be robust or may depend on specific experimental conditions. For example, differences in spike shapes, mostly of the afterhyperpolarization, are often observed between patch-clamp recordings from the same cell type done in different laboratories. This is due to variation in experimental procedures, mainly of the electrode solutions being used. In general it is fairly easy to find examples of experimental observations that could not be reproduced by other groups. Another issue is natural variation among cells (chapter 12, section 3) and in responses of a single neuron over time (Mainen and Sejnowski, 1995; Nowak et al., 1997). Therefore it is important to compile a limited list of experimental observations that should be reproduced, possibly also prioritizing some over others.

In addition, a detailed compartmental model should be, at least at most levels, biophysically correct. These are not phenomenological models like those in chapter 7. For instance, one expects that the same types of voltage-gated channels contribute

to electrical phenomena in the real neuron and the model; it would be awkward if the model used Na^+ channels to generate what in real biology is a Ca^{2+} spike, or vice versa. Because sometimes knowledge about underlying mechanisms is incomplete, it may be valid to use the model to test a hypothesis on a specific biophysical mechanism, but once the facts are known, these should guide the model. This is mainly true for the morphology, ionic channels, and synaptic input. Other aspects are often simulated at a phenomenological level, such as using an exponentially decaying calcium pool to drive channel activation (chapter 4, section 3).

To judge the goodness of a model, it is important to define the accuracy required in advance and also to have reliable ways to validate the model afterward. Currently there are two schools of thought on what is the desired accuracy, based on how the model's parameters are fitted. Modelers who use feature-based fitting methods (chapter 2, section 1) build canonical models that are statistically matched to the data. These models replicate the firing characteristics well, e.g., spike amplitude, firing frequency, and accommodation, because such properties are easy to measure. They often capture subthreshold or bursting behavior less well because this is much more difficult to quantify (examples can be seen in Druckmann et al., 2007; Günay et al., 2008). We have emphasized a more detailed modeling approach, trying to obtain an exact match between the model and specific voltage traces, except for phase differences (chapter 2, section 1), and have demonstrated success at fitting all aspects of complex waveforms (Achard and De Schutter, 2006, 2008; Van Geit et al., 2007). However, this comes at a price because our fitting of channel maximum conductances requires an excellent match of the kinetics of the voltage-gated channels to the data (unpublished observations), which is less the case for the canonical approach. It may also be argued that our approach overfits the data, but the opposite remark can be made about feature-based models. It is our belief that these two views and methods are best seen as complementary. The feature-based approach is excellent for studying natural variability, whether between different neurons (Günay et al., 2008; Prinz et al., 2003) or in the response of a single neuron (Druckmann et al., 2007). It is, however, less suitable for investigating in detail the contribution of specific channels to neural excitability or its modulation and plasticity. The choice may also depend on the experimental data available because, in general, the feature-based approaches require large datasets to allow good statistics. We'll return to the issue of validation at the end of this section.

Data Needed for Constructing a Model

Good active membrane compartmental models are a synthesis of a lot of data about a particular neuron type. The following data are necessary to fully describe the model parameters and, when not measurable, these will instead have to be fitted:

1. the morphology of the neuron:
 - soma,
 - complete dendrite,
 - spine sizes and densities,
 - initial part of the axon.
2. the passive cable parameters.
3. characterization of all ionic channels in the neuron:
 - type and subtype of channels,
 - kinetics for these specific types,
 - channel density at different locations in the neuron,
 - reversal potentials and an indication of whether these are constant or variable.
4. calcium dynamics if calcium-activated channels are present.
5. for synaptic input, if this will be included (see also section 11.4):
 - type and subtype of channels,
 - kinetics for these specific types (including Mg^{2+} block for NMDA channels),
 - density of synaptic connectivity at different locations in the neuron,
 - reversal potentials and an indication of whether these are constant or variable.

In addition, one requires experimental data to fit the model to and for subsequent model validation. It is difficult to give a standard recipe for these data as they will depend on the specific properties of the neuron. Often the voltage response to a series of current steps is used, preferentially evoking several different (firing) response types, if they exist for this neuron type. In addition, simultaneous recordings from the soma and a dendritic location are useful for tuning the dendritic excitability. For example, to fit a Purkinje cell model, we are using paired recordings of the response to current steps from -1 to 2 nA in 0.2 -nA steps. The hyperpolarizing current steps are useful for tuning hyperpolarization-activated channels like the H-current (Roth and Häusser, 2001), while the depolarizing steps first evoke an increase in simple spike rate and then evoke dendritic calcium bursts. Since the neuron fires spontaneously before and after the current steps, additional information is obtained from the transitions. All these data can be obtained from one neuron that is subsequently morphologically reconstructed. One should make sure that the recordings contain as little noise as possible because noise may mask subthreshold behavior.

Many variations are possible; for example, current ramps can also provide quite useful information (Fernandez et al., 2007), as may the activation of specific synaptic inputs (Holmes and Grover, 2006) if this can be controlled well, or white noise

stimulation (mostly used to fit simplified models; e.g., Arsiero et al., 2007; chapter 7). The extracellular potential recorded at a known location relative to the neuron may also provide strong constraints (Gold et al., 2007), but requires computing the model's extracellular potential and possessing exact knowledge of electrode position.

In the near future, electrophysiological data may be complemented or replaced with imaging data. Voltage imaging can in principle be used for effective and easy parameter fits (Huys et al., 2006), while calcium imaging can be used to fit the calcium dynamics (chapter 4), but also to constrain activation of calcium channels. How effectively this, and other types of available data, can be used will in the end depend on the parameter fitting approach used and its flexibility in accommodating different types of criteria or data.

Data Quality

The data listed above are long and extensive. It really helps for the modeler to become completely familiar with all available data. This still involves searching the literature, but is now being complemented with database searches. The most useful web resources in our hands are <http://neuromorpho.org> for neuronal morphologies (chapter 8), the Allen Brain Atlas (<http://www.brain-map.org>) for information on the expression of particular channel subunits, and ModelDB (<http://senselab.med.yale.edu/modeldb/>) for models of channel kinetics. Unfortunately the most important resource, physiological recordings, is not yet available on the web, although initiatives are being taken to improve the situation (Teeters et al., 2008). Whereas in the past the physiological data constraining the model were often derived from published data (De Schutter and Bower, 1994a; Mainen et al., 1995; Migliore et al., 1995), this has become a less acceptable approach. In fact, recent models were usually published together with experimental recordings from the same laboratory (Gasparini et al., 2004; Günay et al., 2008; Watanabe et al., 2002; Solinas et al., 2007).

Even with experimental recordings available, it is not possible to obtain all the necessary data in a single recording, as can be done for passive membrane models (chapter 10, section 6). Sometimes combining data from multiple experiments is desired, as in the feature-based approaches, but when one wants to match the model to a specific experiment, it is inevitable that a lot of supporting data used to characterize the excitability of the neuron and the kinetics of its channels will be based on different experiments. This is a fundamental problem because it is not guaranteed that the neuron or its channels were in exactly the same modulatory state (Park et al., 2006; Jerng et al., 2004) during all these recordings and therefore the data may not perfectly “fit” together.

Quite often not all the necessary data are available for the specific neuron type and animal species studied, and it may not feasible to do the experiments locally. In the

past this has led to the liberal reuse of channel data from models of quite different preparations. The most abused example is the original Hodgkin-Huxley equations (Hodgkin and Huxley, 1952) themselves, which although measured on a squid axon and unable to generate action potentials at vertebrate body temperature, have formed the basis for many models of vertebrate neurons (Traub et al., 1991; Tegnér and Grillner, 1999; Yamada et al., 1998; Destexhe et al., 2001; Wang and Buzsáki, 1996). This problem is partially being overcome by increased sophistication in molecular identification of channels. If one knows that a specific subtype, for example Kv4.2, is present in the neuron being modeled, it is much more acceptable to use kinetic data obtained from Kv4.2 channels present in a different preparation than if one only knows that there is a generic A-type channel (Jerng et al., 2004). In general, computational neuroscientists should start using specific molecular labels and names for channels and proteins because this makes it much easier to communicate our results to systems biologists (Akemann and Knöpfel, 2006; De Schutter, 2008). Future databases of kinetic data for molecular identified channels will facilitate incorporating new channel kinetics into models.

Cell Morphology and Axon

Most of the issues involving neural morphology were covered in previous chapters. Chapter 8, section 1 introduced the methods and challenges in reconstructing morphology and described how to measure shrinkage. Chapter 9, section 2 introduced methods to fix reconstructions that are missing part of the dendrite as a result of the slicing procedure. Chapter 10, section 6 introduced the issue of spine compensation. In practice, many of the problems raised, such as estimating shrinkage or spine densities, often cannot be fully resolved and may have to be settled by parameter fitting.

In compartmental modeling, it is accepted practice to represent the soma with a single compartment, but this must be given an accurate size. Missing in passive models, but quite important in active models, is the axon hillock and initial segment (Mainen et al., 1995). Models that depend exclusively on the soma for initiating action potentials may require unphysiologically high sodium channel densities in the soma (De Schutter and Bower, 1994a), which is not surprising because this is not the physiological site of initiation (Clark et al., 2005; Kole et al., 2008). Usually only the unmyelinated initial axon segment is included in the model, but detailed methods to model myelinated axons with nodes of Ranvier have been developed (McIntyre et al., 2002; Zhou and Chiu, 2001).

Unmyelinated axons impose a large capacitative load on the neuron, which must be included in the model if its c_m is to be correct. An equivalent-cable approach (chapter 10, section 1) can be used to model the axon beyond the spike initiation zone. However, in some cases it may be of interest to model the morphology of the axon in some detail. For example, invertebrate axons often have multiple spike initi-

ation sites. Few detailed axon models exist because it is often assumed that spike conduction can be simply represented as conduction delays (chapter 13, section 3). These models demonstrate how the morphological complexity of axons can influence their filtering properties (Manor et al., 1991) and determine the distribution of time delays (Innocenti et al., 1994).

Channel Parameters

Although it can be a challenge to obtain an accurate morphological reconstruction (chapter 8), the modeler will probably spend more time trying to get the channels right. Once it is decided which channel types to include, the remaining issues are channel kinetics, channel densities, and reversal potentials.

Kinetic data for channels can be difficult to fit correctly (chapter 5, section 6). The experimental recordings may be contaminated by multiple sources of error, such as space clamp errors (Schaefer et al., 2003a; chapter 10, section 3), incomplete pharmacological block of other channels, or voltage shifts caused by ion substitution. If the original voltage-clamp data are available, which is unfortunately rarely the case, it is best to fit the actual traces as opposed to fitting to mean values or curves. In some models, channel kinetics equations based on a particular experiment gave better results than those based on the mean over all experiments, presumably because the voltage dependence of activation and inactivation were correlated in a way that was lost in the averaging (De Schutter and Bower, 1994a). Similar observations have been made for channel densities (Golowasch et al., 2002).

Most channel kinetic data are obtained at room temperature. Will this also be the temperature for the model or will it operate at body temperature? In the latter case, the channel kinetics need to be scaled (equation 5.43) accordingly. Were the channel kinetics obtained at physiological concentrations? If not, it may be possible to estimate the concentration effects on voltage dependence (Hille, 2001; Johnston and Wu, 1995). Is the reversal potential stable or can it shift over time? In the latter case (Yamada et al., 1998), one will need to simulate the ion concentrations and compute the Nernst potential (equation 5.11) or use the GHK equation (equation 5.12), which is the preferred method. Similarly, one may need to compute calcium concentrations to activate calcium-dependent channels. See chapter 4, section 3 for the widely used exponentially decaying calcium pool and more sophisticated alternatives.

A final issue is how to simulate the channel kinetics correctly. Almost all compartmental models use deterministic equations (chapter 5), but there is growing interest in modeling stochastic channel kinetics to investigate the consequences of stochasticity on subthreshold voltage (Steinmetz et al., 2000) and on dendritic (Diba et al., 2006) or axonal (Faisal and Laughlin, 2007) propagation of action potentials. PSICS is a simulator specifically developed for stochastic cable simulation (see the software appendix).

Determining channel density parameters is one of the more difficult parts of model building and usually parameter fitting will play a big role. For the soma, a voltage clamp of acutely dissociated soma can provide good data (Destexhe et al., 1996), but densities in the dendrite or axon are much more difficult to measure. Currently there are four sources of experimental data and all suffer from problems. Counting the numbers of channels in patch-clamp recording, in cell-attached or outside-out mode, has been used extensively to estimate channel densities. Unfortunately, these methods are not reliable (Häusser, 2003) and they can easily be more than a magnitude off in their estimates. This is nicely demonstrated by the long-standing problem of estimating the Na^+ channel density in the pyramidal neuron initial axon segment, first raised by a modeling study (Rapp et al., 1996), which may recently have been finally settled (Kole et al., 2008).

Another quantitative method is a whole-cell voltage clamp, but this is limited by space-clamp problems in the dendrite (chapter 10, section 3). A method has been proposed, however, to correct for space-clamp errors that yields quite accurate measurements for K^+ channels, provided a good morphological reconstruction and estimate of passive parameters are available (Schaefer et al., 2003a). In some cases imaging methods allow accurate measurements, as in dendritic spines (Sabatini and Svoboda, 2000), but for larger structures, most measurements do not provide absolute numbers. The same applies to the final method: quantitative immunocytochemistry, which, whether applied at light or electron microscope level, at best can provide data only on relative distributions of channels.

Fortunately it is not necessary to specify channel densities at the compartmental scale, specifying separate values for $\bar{g}_{i,k}$ for each compartment. Instead, the model can be subdivided into functionally defined zones where channel density values are equal (De Schutter and Bower, 1994a) or some rule specifying the change in density over distance from the soma can be used (Migliore et al., 1999; Holmes and Grover, 2006). These approaches will reduce the number of channel parameters that need to be specified by several orders of magnitude. Similar considerations apply to spine densities, synaptic contacts, etc. However, it should be emphasized that it can be difficult to choose the correct level of granularity of parameters *a priori*, so that different approaches may need to be compared with parameter fitting.

Parameter Setting and Fitting

Once all available data are collected, they can be converted into parameters. It is important to consider which spatial and temporal scales need to be represented accurately. The differential equations (section 11.1) used in these models are not very suitable for simulating events that occur in the μsec range or at the $\text{sub-}\mu\text{m}$ level (see chapter 3), so it does not make much sense to try to reach parameter accuracy at this level. Conversely, very long time constants (in the minute range) may often be

omitted from the model because many other processes in this time range, such as metabolic processes, are not simulated. An important issue in parameterization is standardization. Everything should be in the same units unless the simulator software converts it for you, and should be obtained under the same conditions, e.g., temperature.

Any parameters that cannot be constrained by the data or for which there are reasons to assume that the experimental quantification cannot be fully trusted, will need to be fitted. The art of fitting has evolved a lot in the past decade, with an increasing interest in using automated parameter search methods to fit model parameters. Chapter 2 describes the available methods and approaches in detail; here we will consider some practical issues. Although automated methods offer a lot of advantages, the modeler runs the risk of being more focused on the tuning method than on the model itself. One advantage of hand tuning is that the modeler develops a feeling for the model's properties and its parameters' sensitivities during the parameter fitting.

The most important thing is to realize that automated parameter search methods are evolving rapidly and it may be useful to check recent literature or web resources to get an update on the current state. Invest effort in developing a good fitness measure (chapter 2, section 1) for the model because this is the most critical component of the process. This may require prototyping on test runs because the fitness measure depends on the quality and expanse of the available data. As already mentioned, the fitting method used will also determine how closely the model is fitted to a particular dataset, which, depending on the scientific questions asked, may be important or not. Finally, these methods often do not generate unique solutions but instead produce multiple good models (Major et al., 1994; Achard and De Schutter, 2006, 2008; Holmes and Grover, 2006). This allows the generation of families of models (chapter 12, section 3), which opens interesting opportunities for studying homeostasis (Marder and Goaillard, 2006; Achard and De Schutter, 2008) or populating network models.

What should be the strategy to fit disjointed sets of parameters? A classic approach has been to fit the passive responses first, as a kind of skeleton for the model (Segev and Burke, 1998), and then add the voltage-gated channels and fit their densities. However, doing a good passive fit assumes experimental data from a neuron in which all voltage-gated activity is blocked and that also excludes recording spiking activity under different conditions in the same neuron. Though one could in theory first record the active data and then the passive, this would cause a very long recording session, leading to morphological artifacts like blebbing of dendrites (Jaeger, 2001; chapter 8, section 1). So it may be a better strategy to obtain sufficient information about passive properties by using hyperpolarizing current steps with all channels (including H-current, I_h) active and immediately fit an active model (Holmes and

Grover, 2006). We have had good success using Neurofitter (see the software appendix) to fit Purkinje cell data, where we fitted at the same time C_m , R_a , spine density (using sizes from the literature; Harris and Stevens, 1988) and maximum conductances for the leak current and I_h (unpublished data). However, this worked only after shifting the voltage dependence of the I_h kinetics by hand because with the original equations (Akemann and Knöpfel, 2006) I_h did not activate sufficiently. Our experience with Neurofitter suggests that close fitting of actual voltage traces works only if the channel kinetics match the experimental recording very closely. We are now exploring the inclusion of automated adaptation of the channel kinetics to the parameter search.

Validating the Model

It is important to validate the resulting model, whether it was tuned manually or automatically. As described in chapter 12, section 1, it is instructive to perform a parameter range study at the end of the tuning process. Such a study cannot validate the model, but by giving insights into the important parameters, it may indicate potential problems. For example, the size of the leak conductance should not be a critical parameter; if it is, your model is probably missing one or more potassium channels. Similarly, the leak reversal potential should not deviate too far from the resting membrane potential. A parameter range study will also provide insight into the uniqueness of the model. In the best case, the parameter range study may point to simple experiments to better constrain the model; in the worst, it may indicate a fundamental flaw.

In the older literature it was considered important that a model be robust to large changes in parameter values. This is not a realistic requirement for active membrane models because it may be important for a neuron to keep its operational range close to a bifurcation point, such as a switch from regular firing to bursting, which may be possible only in a tight parameter range. In biological neurons, many homeostatic processes continually adapt channel properties and densities, and these may be employed by nature to keep some parameters in a small range, although a number of such ranges may coexist (Achard and De Schutter, 2006; Marder and Goaillard, 2006).

The only way to validate a model is to test whether it can reproduce behaviors for which it was not tuned. It is best to plan for this before the tuning process and, following the example from the machine learning community, split the available dataset into a tuning set and a validation set. This strategy has recently been successful in fitting integrate-and-fire models (chapter 7) to data from dynamic-clamp experiments (chapter 12) (Badel et al., 2008). Again, the best way to do this depends on the data available. If one is tuning a fairly large number of parameters, it is important to make the range of behaviors covered by the tuning dataset as rich as possible, which

may leave the validation dataset rather trivial, like an extra current step. If one can design the experiments used for the fitting, it is worthwhile to add specific experiments for the validation set, such as a current ramp.

We have found it useful to use a validation set that is qualitatively different from the tuning set. The reference Purkinje cell model (De Schutter and Bower, 1994a) was tuned to reproduce the responses to intracellular injection of current *in vitro* (based on the literature). Subsequently it was validated by showing that the same model could also simulate responses to synaptic activation of four different kinds of inputs both *in vitro* and *in vivo* (De Schutter and Bower, 1994b). Although the latter required the tuning of a few additional free parameters (the \bar{g}_{syn}), it indicated that the basic active membrane model remained faithful to the biology under widely different circumstances.

11.4 Running the Model

Model Initialization

Many neurons fire spontaneously, which implies that active membrane models that simulate this behavior will have state variables that are never at equilibrium. More specifically, it is impossible to analytically predict the proper values for channel gating variables like m and h (equation 11.2) or for the calcium concentrations for a particular value of V . The only way to obtain these values is by numerical simulation. It is not trivial to decide at what voltage to initialize the model, and then it may take a long equilibration time before its behavior settles to some steady state, for example, a fixed firing rate. If the model is bistable, its “spontaneous” behavior may depend on how it was initialized.

For a spontaneously firing model, it is best to initialize at a voltage about midway in the interspike interval and set channel gating variables to the corresponding equilibrium value. As pointed out, these values are approximations and the model will need to equilibrate before it reaches steady state. Calcium concentrations are more difficult to predict and it may be useful to try out different values because in some neurons they may be well above resting concentration.

Notice that the initialization issue also affects tuning of the model. Therefore it may be easier to fit data where the neuron is slightly hyperpolarized before the current step to suppress firing, although such data will also be less informative about transitions in firing activity. If one tunes a spontaneously firing model, then it is important to allow for model equilibration before measuring the fitness of the model output.

Once the model is tuned, however, and if one plans a lot of simulations under similar conditions, then it is worthwhile to save a snapshot of the fully equilibrated

model and use that to initialize the state variables. This will (largely) eliminate the simulation time needed to equilibrate.

Simulating Synaptic Input: Spines

In practice, most of the data used for model tuning are obtained in a slice where synaptic input is absent or sparse. So in many cases the model tuning is done in the absence of synaptic input, and this is only added afterward during simulation.

A large part of the membrane surface of most neurons consists of spines—over 50% in the case of Purkinje cells (Rapp et al., 1992). It is not possible to measure these faithfully using light microscopy (chapter 8, section 1) and omitting them will lead to large errors in the fitted cable parameters, typically leading to unphysiologically large values for C_m and low ones for R_m . Spines contribute nothing to the firing properties of neurons in the absence of synaptic input and may be expensive to simulate (at least two compartments per spine, one for the neck and one for the head), so they are often not modeled explicitly. Methods for spine compensation are discussed in chapter 10, section 6, where it is also mentioned that one needs to decide whether one wants the spines to be active or passive. While electron microscopy can provide data about spine sizes and local densities, their actual number on a neuron is almost always unknown and is therefore one of the parameters that needs to be fitted (section 11.3).

Opinions are divided on whether to explicitly model spines when excitatory input is added to the model. We have chosen to do so in the Purkinje cell model (De Schutter and Bower, 1994b), while Poirazi et al. (2003) did not do so in pyramidal neuron models. The main effect of spines is a voltage attenuation from the site of synaptic contact with the dendrite (chapter 10, section 3), which can be mimicked by reducing the synaptic conductance. Nevertheless, as long as one does not want to simulate all the spines explicitly, it makes the model biophysically more realistic to include them explicitly at sites of synaptic input; this also allows for the inclusion of special mechanisms localized to spines, such as the calcium required for induction of synaptic plasticity (Holmes and Levy, 1990). In the Purkinje cell model, we decided not to model all 150,000 parallel fiber inputs (Harvey and Napper, 1991), but instead only about 1% of them (see next). This implies that we modeled 99% of the (passive) spines implicitly and 1% explicitly, using the following equation to replace equation (11.4):

$$s'_i = \pi d_i l_i + s_{sp}(f_{sp}l_i - N_{sp}) \quad (11.10)$$

where s_{sp} is the mean surface area of the spines (Harris and Stevens, 1988), which matches that of the explicitly modeled two-compartment spines, f_{sp} is the spine density (per unit of length), and N_{sp} is the number of explicitly modeled spines per compartment (usually one).

Simulating Synaptic Input: Number of Inputs and Their Activation

Here we briefly consider how to provide synaptic input to an isolated cell model; see chapter 13, section 3 for similar issues in network models.

Two types of synaptic input can be distinguished and, at the level of parametrization, they should be treated differently. First there is the “background” synaptic input used to mimic the continuous bombardment by synaptic input that many neurons receive *in vivo* (Destexhe and Pare, 1999). This can be complemented with patterned input, usually of a synchronous nature.

In many neuron models, the background synaptic input causes firing of action potentials. One can mimic this by adding a constant synaptic conductance to the membrane (Rall, 1964 and equations 13.13–13.14), but this will cause regular firing. Stochastic firing can be achieved by simulating a balance of excitatory and inhibitory inputs (De Schutter and Bower, 1994b; Jaeger et al., 1997). In a balanced inhibition model, excitatory and inhibitory synaptic currents cancel each other out and irregular firing is driven by the small fluctuations around zero (Shadlen and Newsome, 1994; Troyer and Miller, 1997; Salinas and Sejnowski, 2000). Note that only the synaptic currents matter; the conductances can be quite different. As a consequence, the simulated synaptic input works as a distributed voltage clamp. It drives the neuron to the membrane voltage where the sum of the currents is zero and counters deviations from this voltage (Jaeger et al., 1997). These mechanisms can be mimicked experimentally by using dynamic-clamp procedures (Jaeger and Bower, 1999; Miura et al., 2007; chapter 12, section 4), with the important difference that the synaptic current is now applied at a single location in the neuron. Alternatively, analysis of the power spectrum of synaptic noise in real neurons can provide constraints for the kinetic models of synaptic currents (Destexhe and Rudolph, 2004).

How many synaptic contacts should be modeled to simulate background activation? Large neurons receive inputs in the tens of thousands to hundreds of thousands, but is it necessary to simulate them all? Even though computer memory and speed now allow it, it does not make much sense to model all synapses. First, the number of compartments modeled is usually much smaller, so that 10–100 synaptic contacts would have the same isopotential postsynaptic target. Second, the background input will be activated by generation of random numbers and it makes little sense to do this for all synapses. For conditions of little synaptic saturation that is caused by temporal summation (small \bar{g}_{syn} and low activation rates f_{syn}), the total, distributed synaptic drive A onto the cell approximates

$$A \approx N_{\text{syn}} \bar{g}_{\text{syn}} f_{\text{syn}}. \quad (11.11)$$

This relation suggests that one can compensate for a reduction in the number of synapses simulated (N_{syn}) by increasing \bar{g}_{syn} or f_{syn} . For the Purkinje cell model, we have shown that interspike interval distributions remain constant over a range of

N_{syn} values (De Schutter and Bower, 1994b) and that the effect of learning on spike output is statistically independent of N_{syn} (for values larger than 1,500; Steuber et al., 2007). Therefore, within the limit of low \bar{g}_{syn} and low f_{syn} , equation (11.11) can be used to change the number of synaptic contacts as desired or to estimate how the synaptic activation rates in the model relate to physiological ones. Note that equation (11.11) holds only for asynchronous activation of synapses: the N_{syn} should be activated independently of each other. This requires a good quality of random number generation because long-range correlations may introduce unwarranted output correlations (chapter 2, section 7). In general, N_{syn} will be set so that for each type of input there is a single synaptic contact on any compartment.

If one wants to simulate repeated runs with different realizations of the background synaptic input to generate a peristimulus histogram (PSH; De Schutter and Bower, 1994c) the following trick will be useful. Owing to identical initial conditions (see earlier discussion on equilibration), the first spikes fired by the neuron model will tend to occur close to fixed times, even with different background synaptic inputs, leading to oscillatory artifacts in the PSH. An easy way to counter this is to use a random, short run time that is discarded before recording the actual simulated traces used to generate the PSH.

In addition to the background synaptic input, patterned input can be applied, usually in the form of a synchronous activation of synapses. The interaction between these two forms of input is described as gain modulation (Chance et al., 2002; Fellous et al., 2003; Prescott and De Koninck, 2003). In the case of synchronous activation, there is no firing frequency and there is no advantage in modeling more than one synaptic input per compartment, with the appropriate value for \bar{g}_{syn} . An alternative approach is to introduce correlations in the background synaptic input (Destexhe and Pare, 1999).

11.5 Future Directions

As already pointed out in the overview in section 11.3, surprisingly few detailed active compartmental neuron models have been published recently. One reason may be that even with automated parameter-fitting methods, creating such a model is a lot of work and may take a frustratingly long time. On the other hand, maybe the field has also not been very forward-looking and is stuck too much in the classic compartmental model plus Hodgkin-Huxley paradigm. Computational neuroscience has not kept up with the rapid developments in systems biology (De Schutter, 2008), which suggests many ways in which complex neuron models could be improved and made more interesting. For example, the continued use of the exponentially decaying calcium pool as the standard model for calcium dynamics is really out of touch with

the sophisticated, quantitative models of calcium buffering that are now available (Schmidt et al., 2003; chapter 4, section 3). Shouldn't we explicitly model the ionic homeostasis in the neuron, including the different pumps and exchangers (De Schutter and Smolen, 1998)? If, as has been proposed by many experimental and modeling studies, homeostatic mechanisms in neurons really control the firing mechanisms instead of setting specific sets of channel densities (Marder and Goaillard, 2006), wouldn't it be better to develop a class of homeostatic neuron models and skip most of the parameter fitting mentioned here? Since such models require a representation of the signaling pathways responsible for homeostatic control processes (Liu et al., 1998), they would be closer to molecular modeling of neurons at the cellular level. Based on these considerations, we predict a bright future for complex neuron models that add molecular details to the classic mix of morphology and channels.

12 Realistic Modeling of Small Neuronal Networks

Ronald L. Calabrese and Astrid A. Prinz

Small networks of neurons are attractive objects for modeling using conductance-based or “realistic” neuron models (De Schutter et al., 2005; Grillner et al., 2005; Marder et al., 2005). This is particularly true for invertebrate preparations with identified neurons. Often in invertebrate networks, the membrane currents of the component neurons and their synaptic interactions can be experimentally characterized using a voltage clamp, so that sensible conductance-based model neurons can be constructed. Small network models, however, are not limited to invertebrate networks but can be starting points for modeling large vertebrate networks that are modular in their wiring diagram. Importantly, for small networks it is often possible to precisely quantify the network behavior that one wishes to simulate so that a benchmark for a model’s performance can be established. This chapter builds on the cellular modeling presented in previous chapters to explore methods of modeling small networks of neurons. The neuronal elements modeled can be dynamically complex in terms of the content of voltage-gated currents (chapter 5) and the properties of synaptic inputs and outputs (chapter 6), but in such network models they are typically simplified morphologically for computational efficiency, because their morphology has not been quantitatively described, and because of the complexity of establishing conductance distributions throughout a neuronal structure. These models have been successful as experimental tools, where experiments lead to construction of models and models lead to new testable, experimental predictions in a beneficial cycle.

12.1 Basic Modeling Strategy for Small Networks

Small networks afford the opportunity to use realistic conductance-based modeling techniques (described in chapter 5) without straining the power and speed of current computers. Moreover, there are software packages available to facilitate such modeling (see section 12.6). Nevertheless, there are cases where “point” neuron models such as simple integrate-and-fire models or hybrid models containing some voltage-gated conductances or other mathematical “reductions” (described in chapter 7) can

be used effectively, especially when details of biophysical analysis of active membrane conductances are not available (see Kleinfeld and Sompolinsky, 1988 for an interesting example). Even with extensive biophysical analysis of network neurons, compromises still have to be made: how much detail is incorporated into any particular network model and how biologically accurate the detail is, are critical. Not enough detail will lead to false conclusions, whereas too much detail wastes valuable time and can divert attention from the significant parameters that affect network function.

The ultimate consideration in deciding the level and accuracy of the detail incorporated is the purpose for which the model is being made. If one is seeking to build an experimental model that can organize accumulated data and guide future experiments through model prediction, then the level of detail and its accuracy should be as high as possible. If on the other hand, one wishes to make a more heuristic model that maximizes generality to other networks, the addition of idiosyncratic details should be minimized and the accuracy of the detail described may be compromised to incorporate parameters that are more transparent or are functionally, rather than biophysically, based. The choices that the researcher makes about the type of model he or she wishes to build and thus what is included and what is excluded will determine the usefulness of the model both as an experimental tool and in terms of the insights into basic biological mechanisms that it can afford.

Making the Model

Once one determines the type of model one wants, then implementing that vision can be straightforward, especially given the large number of neuronal simulation environments available (see section 12.6). The realistic (conductance-based) modeling of small networks boils down to solving the current balance equation for any single-compartment neuron model or any compartment within a neuron model:

$$c \frac{dV}{dt} = - \sum I_{\text{ion}} - \sum I_{\text{syn}} - \sum I_{\text{inject}}, \quad (12.1)$$

where I_{ion} represents intrinsic membrane currents, I_{syn} synaptic currents, and I_{inject} injected currents (or the current from connected compartments). Many of the details and accuracies of which we wrote earlier concern the compartmental structure of the neuron models of the network, and the details of the equations that describe I_{ion} and I_{syn} .

Neuronal Morphology

The full morphological detail of neurons (see chapters 8–11) is almost never implemented in small network models; rather, simplifications are made to increase compu-

tational efficiency and to focus attention on circuit interactions. In our own work modeling half-center oscillators (pairs of mutually inhibitory interneurons) that form the timing kernel of the generator network of the leech heartbeat motor pattern, we at first chose to ignore the morphological complexity of the oscillator interneurons and treat each cell as a single isopotential compartment (A. A. Hill et al., 2001). This simplifying assumption was made for several reasons. (1) There were a large number of voltage-gated and synaptic currents that experimental evidence demanded be incorporated into each model neuron, which called for computational efficiency. (2) The available data suggested that these neurons are electrically compact and receive a limited number of inputs with widely distributed synaptic contacts. (3) We lacked detailed knowledge of the distribution of voltage-gated and synaptic conductances within the neurons. Similar considerations have often caused other workers modeling networks generating motor patterns to use model neurons with one or a few compartments (De Schutter et al., 2005).

Ultimately, the level of morphological detail included must depend on both the available data and the scientific questions asked of the model. If there is evidence, for example, of segregation of electrical activity within neurons that influences circuit behavior, then morphological detail adequate to accommodate such segregation must be incorporated (see section 12.2). Nevertheless, it may be beneficial to start by using neurons comprising only a few compartments (see chapter 11, section 2), with the view that additional morphological complexity can be added as data become available and when comparison with experimental data reveals model discrepancies that may have a morphological basis.

Intrinsic Membrane Currents

The important intrinsic membrane properties of the component neurons that form the network must be described mathematically, most likely with classical Hodgkin-Huxley-type equations (Hodgkin and Huxley, 1952) based on voltage-clamp records of individual ionic currents (see chapter 5, section 2). Markov models for individual ionic (and synaptic) currents can be applied to neuron models (see chapter 5, section 5) and may prove useful in network simulations in the future. In some cases, voltage-clamp data may not be available for all cells or all currents in particular cells, and current equations must be bootstrapped from current-clamp recordings. It is then efficient to start from an existing mathematical description of a related current or the same current from a related neuron. For example, it is often the case in invertebrate neuronal networks that no data on the fast Na current (I_{Na}) that mediates spikes is available because of space-clamp limitations of voltage clamps applied to the cell body. In our own work, our solution to this problem was to adapt the fast Na current from the squid (i.e., the original Hodgkin-Huxley equations for I_{Na}) to fit the spikes observed in leech heart interneurons.

In several preparations there are particular neurons for which accurate mathematical descriptions of voltage-gated currents are available that can serve as bootstrapping resources for modeling voltage-gated currents in neurons for which voltage-clamp data are unobtainable or fragmentary. For example, for the stomatogastric neuronal networks of crustaceans, the crab LP neuron serves this role (Buchholtz et al., 1992; Golowasch et al., 1992). Often these current descriptions are incorporated into libraries in simulation packages such as NEURON or GENESIS (see the software appendix) and are easily modified to bootstrap related currents.

Synaptic Connections and Electrical Coupling

Chapter 6 presents methods for modeling synaptic interactions in network models and discusses the strengths and weaknesses that determine the choice of one method over another. Ultimately, the data available on a particular synaptic connection will dictate the detail that can be incorporated into its description. Often these data are limited to current-clamp measurements that do not justify going beyond the simplest descriptions for spike-mediated synaptic transmission. In many invertebrate generator networks for motor patterns, graded transmission occurs concomitantly with spike-mediated transmission, and special mathematical descriptions or hybrid descriptions have proved useful; these are especially informative when they are based on biophysical experimental analysis (De Schutter et al., 1993; Nadim and Manor, 2000; A. A. Hill et al., 2001; Mamiya et al., 2003).

Electrical coupling through gap junctions can also play a role in determining the electrical activity of small networks and must be included. In its simplest form, non-rectifying electrical coupling can be expressed by

$$I_{\text{elec}\,1} = g_{\text{elec}}(V_1 - V_2), \quad I_{\text{elec}\,2} = -I_{\text{elec}\,1}, \quad (12.2)$$

where V_1 is the membrane potential of the receiving cell (cell 1) and V_2 is the membrane potential of the coupled cell, cell 2. Rectification is handled by conditional statements, e.g., specifying that if $(V_1 - V_2) > 0$, then $I_{\text{elec}} = 0$ will make an electrical synapse rectifying from cell 2 to cell 1. Low-pass filtering associated with coupling sites electrotonically distant from integration or recording sites is accomplished by low-pass filtering of V_1 and V_2 .

Parameterization

Once a model has been constructed, the selection of model parameters is of paramount importance. These parameters include the values that determine the voltage dependence of time constants and activation variables for voltage-gated currents, and the maximal conductances and reversal potentials for all currents. The selection of parameters can be thought of as tuning to experimental data, which in principle seems simple but in practice is difficult. Much of the data on intrinsic membrane cur-

rents and synaptic interactions that are incorporated into a model will be generated in biophysical experiments under artificial conditions, but the network model will ultimately be compared with current-clamp records during normal physiological activity. Moreover, circuit output in the living system is often variable; e.g., in central pattern generators (CPGs) period can vary over a considerable range (twofold or more; e.g., Bucher et al., 2005; Norris et al., 2006). Deciding whether to tune the model to average circuit characteristics or to a specific typical example can be difficult. On the other hand, a properly tuned model must be able to capture observed biological variability by physiologically appropriate variation in parameters. Viewed in this way, biological variation can act as a constraint on model parameters. Ultimately, the procedures for establishing whether the model conforms to the observed behavior of the living network are of necessity ad hoc and dictated by the experimental question the model is to address.

First, as many model parameters as possible should be fully specified by the data, for example, by fitting voltage-clamp data. This process will inevitably leave some parameters unknown or unconstrained. Moreover, there is now a growing awareness that the maximal conductances of intrinsic and synaptic currents vary significantly (twofold or more) in the living system without altering characteristic network output (Marder and Goaillard, 2006; Schulz et al., 2006). Furthermore, using average values of maximal conductances to parameterize a model can lead to failure to capture neuronal or network activity (Golowasch et al., 2002). On the other hand, this realization should not be taken to mean that biological variation in maximal conductances is not reflected in aspects of oscillatory network output, such as period and phase.

Once a modeler has constrained parameters as the data warrant, the free parameters must be tuned to produce realistic cellular and network activity. Chapter 2 describes methods that can be used to find parameter sets that will work; ultimately these methods return parameter sets as good as the fitness function used and the benchmark activity upon which that fitness function is based. Section 12.3 describes database methods that can be used to define the “totality” of free parameter space that supports observed network activity.

Defining a Canonical Model

The parameterization methods discussed here along with conventional hand tuning can be used to define a canonical or benchmark model that represents a parameter snapshot of the living network to be used in an experimental context. Essentially, those who wish to derive mechanistic and experimental insight must choose a reference or starting point that reflects the present understanding of the living network. The canonical model does not represent the optimal description of the living network; indeed such a single description may not exist (see section 12.3), but the aim

of such a model should be to capture a description that can be used to generate experimentally testable hypotheses about the network.

In selecting a canonical model, it is important to determine that it is robust, reflecting the robustness of the living system, which owing to biological variation, can show variation in both network activity and underlying free parameters. Practically, this robustness means that the model is not overly sensitive to the choice of particular values for free parameters, and with biologically reasonable variation of relevant free parameters, can express a range of network activity observed in the living system. It is possible that the values of the chosen parameters are close to a bifurcation in system behavior: a point in parameter space where system behavior splits along different mechanisms of cellular or network operation (see e.g., Cymbalyuk et al., 2002). Alternatively, the model might be extraordinarily sensitive to a particular parameter so that careful measurement of that parameter must be undertaken to constrain it better. Only by a systematic study of the model through parameter variation, including where feasible, database approaches, can one be assured that a given canonical model is robust enough to serve as a testbed for developing a hypothesis.

Once a canonical model is chosen, sensitivity analysis can be used to identify parameters critical to proper network function or control of key aspects of network output. In performing sensitivity analyses, it is beneficial to focus on measures of network output that reflect these biological priorities and to determine the sensitivity of these measures to variation in free parameters. For example, in rhythmically active CPGs, such measures might include phase, which may be limited to a narrow range in the living system for proper function, and period, which can be expected to vary smoothly over a physiologically relevant parameter range. It is then useful to determine the sensitivity of the canonical model's output variable to a small change in each parameter from the canonical value. For example, in a model of the interaction of the pyloric and gastric networks in the crustacean stomatogastric ganglion (Nadim et al., 1998), maximal conductances of all synaptic currents (and certain voltage-gated currents) were free parameters and the output measure used was gastric period. Parameter changes of +10% from canonical values were imposed and sensitivity was assayed as

$$S_{\text{output}} = \frac{\Delta \text{output}/\text{output}}{\Delta \text{parameter}/\text{parameter}}. \quad (12.3)$$

This analysis showed that the model's period was not very sensitive to most parameters (the absolute value of $S_{\text{period}} < 1.25$) but identified a key synapse that was important in determining the model's period. Large parameter sweeps then could be used to assay the role of this synapse in controlling the model's period.

It is then helpful to explore parameter space with the canonical model as a starting point by varying each free parameter $\pm 10\%$ and assessing the effect on output mea-

sures of choice. In a CPG network, for example, there may be regions of parameter space where rhythmicity becomes erratic or breaks down, whereas in other regions period remains constant or varies smoothly. Only with systematic sensitivity analysis (Olypher and Calabrese, 2007; Weaver and Wearne, 2008) or database approaches (Prinz et al., 2003, 2004a) can one quantitatively determine how such parameter variations interact. Nevertheless, individual parameter sweeps in conjunction with monitoring intrinsic and synaptic currents and conductances (or their activation and inactivation variables) and close observation of physiologically relevant aspects of network output are invaluable for gaining insight into the mechanisms of network function and for generating testable hypotheses with a model. The order in which one performs these assays—parameter sweeps and sensitivity analysis—is arbitrary and the process is often iterative.

Model-Generated Hypotheses

An advantage of experimentally based realistic network modeling is that it can interact directly with biological experiments. Parameter analysis and database approaches (see section 12.3) identify key parameters that influence network function and generate experimentally testable predictions. To close the interactive loop, these hypotheses must be tested, the aim being refinement of our mechanistic understanding of the living system and the identification of shortcomings in the canonical model so it may be improved. In small networks, e.g., it is often possible to alter key parameters such as \bar{g} of voltage-gated and/or synaptic currents using dynamic current clamping (Prinz et al., 2004b; Goaillard and Marder, 2006) and thus directly test model predictions generated during parameter analysis (see section 12.4). Alternatively, voltage clamping with realistic voltage waveforms can be used to directly test current flows predicted in the model (e.g., Olsen and Calabrese, 1996). In the interaction between model and experiment, “more is better” is a reasonable rule of thumb, and the experimentalist and modeler (often, it is hoped, the same person) must work together to generate experimentally feasible predictions and identify changes needed when a model’s predictions fail in experiment. The experimental failure of a model’s prediction is a critical point in its development, indicating that the model’s parameters, assumptions, and/or structure require new scrutiny and thinking. In this sense an experimentally based network model is a means to help one understand network function and not in itself an end; modeling is thus an iterative process: experiment, model, experiment, model, experiment

Model Expansion and Compromises

In modeling a small neuronal network, one should take a step-by-step approach. Most experimentalists have a sense of how “their” network functions, and these hunches can help in model development. One may start by identifying key elements

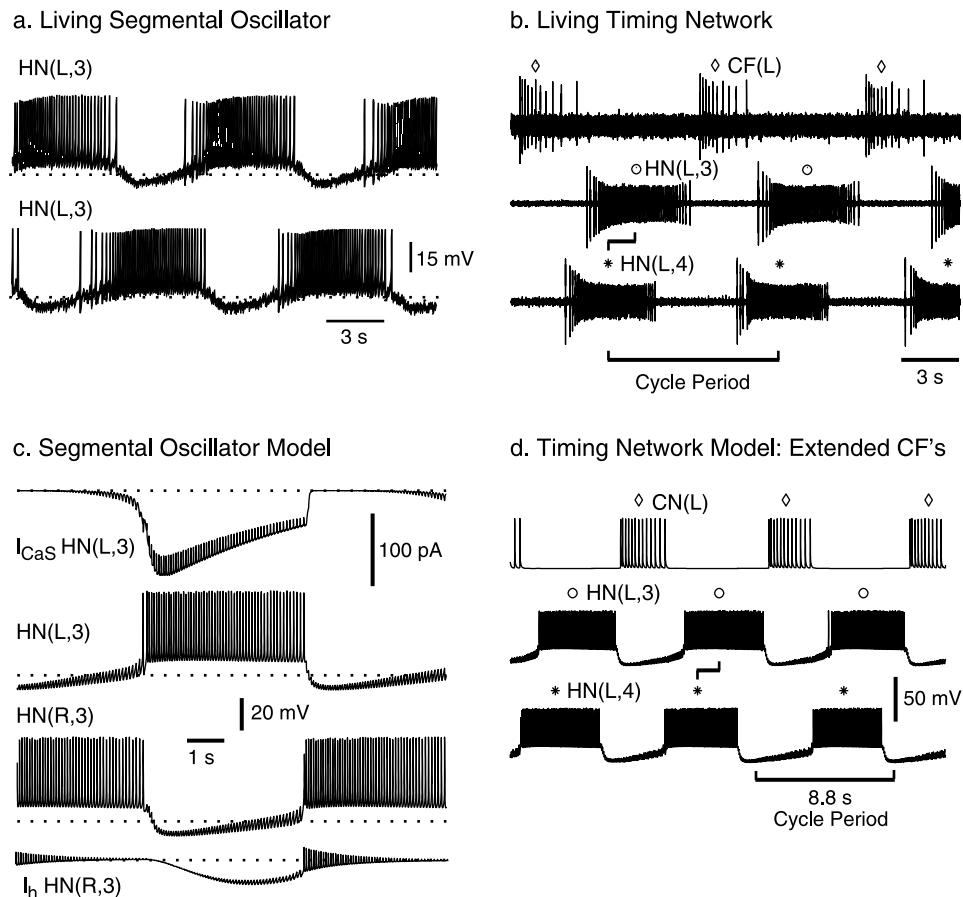
in the network and focusing initial modeling attempts on them. For these key elements, strong experimentally based and thoroughly tested cellular models are necessary; for other elements, such rigorous representations may not be necessary at first. For example, in the work on the interaction of the pyloric and gastric mill networks (Bartos et al., 1999), the entire pyloric circuit was represented as a rhythmic synaptic input to key gastric model neurons. This simplification was tenable because the aim of the model was to explore how pyloric input influences gastric period. Thus feedback from the gastric circuit to the pyloric circuit was not relevant. This simplification increased computational efficiency and focused attention on the pertinent issues. Let the model fit the question; detail for the sake of detail can impede progress, but all the details necessary to answer the questions asked of the model must be included.

12.2 Leech Heartbeat Timing Network: A Case Study

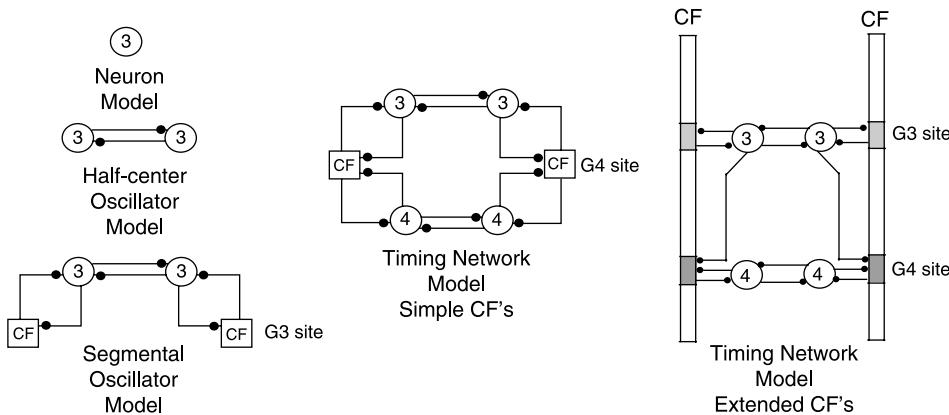
To illustrate the modeling process described, we will follow an example from our own work on motor pattern generation. There is a rich literature on modeling small networks that is filled with other examples to which the reader may gain access by some recent reviews (De Schutter et al., 2005; Marder et al., 2005). The essence of our approach has been the close interplay of modeling and experiment, so that insights derived from experiments are used to improve the model and the model's prediction guides the choice of experiment.

Experimental Background

A network of seven bilateral pairs of segmental heart interneurons produces rhythmic activity that paces segmental heart motor neurons, which in turn drive the two hearts of the leech (see Calabrese, 1995 and Kristan et al., 2005 for reviews). The synaptic connections among the heart interneurons and from the heart interneurons to the motor neurons are inhibitory; electrical coupling between interneurons and between motor neurons in the same segment is also present. The first four pairs of heart interneurons can reset and entrain the rhythm of the entire pattern-generating network of interneurons. Thus this network is designated the *timing network* (eight-cell network) for heartbeat. The other three pairs of heart interneurons are followers of these anterior pairs. Two foci of oscillation in the timing network have been identified in the third and fourth ganglia, where the oscillation is dominated by the reciprocal interactions of the third and fourth pair of heart interneurons, respectively. Mutually inhibitory synapses between the bilateral pairs of heart interneurons in these ganglia (figure 12.1a), combined with an ability of these neurons to escape from inhibition and begin firing, pace the oscillation. Thus each of these two mutually inhibitory heart interneuron pairs can be considered an elemental *half-center*

**Figure 12.1**

Biology and modeling of the leech heartbeat timing network. (a) Simultaneous intracellular recordings showing the normal rhythmic activity of two reciprocally inhibitory heart (HN) interneurons that compose a segmental oscillator in an isolated ganglion preparation. Heart interneurons are indexed by body side (R, L) and ganglion number. Dashed lines indicate -50 mV in voltage traces. (b) The electrical activity of three ipsilateral heart interneurons recorded extracellularly from a chain of ganglia. The oscillator heart interneurons are labeled and indexed as in (a) and the coordinating fiber as CF with a body side index. The G4 [HN(L,4)] oscillator interneuron leads the G3 [HN(L,3)] oscillator interneuron in phase, and the coordinating fibers fire in alternation with both oscillator interneurons. Markers \diamond , \circ , $*$ indicate the midpoints of each burst for the CF, HN(3), and HN(4) interneurons, respectively, to facilitate the visualization of the cycle period and the HN(3) to HN(4) phase lag. (c) Segmental oscillator (six-cell) model of electrical activity and calculated slow calcium current (I_{CaS}), and hyperpolarization-activated current (I_h) in opposing model oscillator interneurons (coordinating neuron activity not shown). Dotted lines indicate -50 mV in voltage traces and 0 nA in current traces. I_h is slowly activated by the hyperpolarization caused by synaptic inhibition, adding a delayed inward current that promotes escape and drives activation of I_{CaS} , which in turn supports burst formation. During the burst, I_{CaS} inactivates, causing spike frequency to wane slowly, reducing inhibition to the opposite cell and promoting its release. (d) Activity of the timing network model with extended coordinating fibers. Symbols and labels as in (a) and (b).

**Figure 12.2**

Schematic circuit diagrams of different models. The figure shows the steps from the development of a neuron model of an oscillator heart interneuron to a timing network model with extended coordinating fibers (CFs). Open circles represent cell bodies, open boxes represent sites of spike initiation, and small filled circles represent inhibitory synapses. Shown from top left to right, single G3 oscillator neuron model, (two-cell) half-center oscillator model for G3, (six-cell) segmental oscillator model for G3 (CF stumps represented as a single compartment), timing network model with simple CFs represented as a single compartment in G4, and timing network model with extended CFs. Only one CF is shown on each side for clarity. The CFs were modeled as multicompartamental cylindrical cables. Each fiber had two spike initiation sites (shaded boxes) that were each capable of spontaneous activity, one in ganglion 3 (G3 site) and one in ganglion 4 (G4 site). The G3 and G4 initiation sites, each consisting of a group of five compartments, were separated by fifty conduction compartments representing the interganglion portion of the fiber. Both sites showed postinhibitory rebound, and the G3 site fired at a higher average frequency than the G4 site (indicated by deeper shading). See Jezzini et al. (2004) for details.

oscillator and these interneurons are called oscillator interneurons (figure 12.2). The heart interneurons of the first and second ganglia act as coordinating interneurons, serving to couple these two elemental oscillators, thus forming the (eight-cell) timing network for the system (figure 12.2). When the third and fourth ganglia are separated by cutting the connective, each isolated third and fourth ganglion contains a (six-cell) *segmental oscillator* that consists of a pair of reciprocally inhibitory oscillator interneurons (half-center oscillator) and the active “stumps” of the coordinating neurons, which provide additional inhibition (figures 12.1a and 12.2). Under normal conditions, the coupling between the segmental oscillators causes the activity of the G3 oscillator to lag that of the G4 oscillator on the same side by about 5% in phase (figure 12.1b).

Coupling between segmental oscillators by the coordinating neurons is quite strong and the segmental oscillators are mutually entraining (Masino and Calabrese, 2002b,c). The phase lag between the segmental oscillators is a function of the entrainment conditions. For example, when a G3 pair entrains a G4 pair to a shorter period, the driving pair (G3 interneurons) lags behind the driven pair (Masino and

Calabrese, 2002a). Conversely, when one pair entrains the other to a longer period, the driving pair leads.

Several ionic currents have been identified in single-electrode voltage-clamp studies that contribute to the activity of oscillator heart interneurons (Calabrese, 1995). These include a slowly inactivating low-threshold Ca current (I_{Cas}), a hyperpolarization-activated inward current (I_h), and a leak current (I_{leak}). The inhibition between oscillator interneurons is primarily though not exclusively (graded transmission is also observed) spike mediated and shows short-term plasticity during each cycle (Olsen and Calabrese, 1996; Ivanov and Calabrese, 2003). This wealth of experimental analysis demanded modeling to organize the plethora of data and facilitate insights into the interactions between membrane properties and synaptic connections that determine network function and that can guide future experimental exploration.

Oscillator Heart Interneuron Model

Good network models always start with good neuron models (figure 12.2), and the availability of extensive voltage-clamp data for oscillator interneurons made conductance-based modeling an obvious choice. The first step in the modeling process was the development of standard Hodgkin-Huxley representations of each voltage-gated current using methods like those described in chapter 5. Synaptic transmission between the oscillator neurons is complex, involving spike-mediated transmission and graded transmission, that show dynamics based on membrane potential in the former and low-threshold Ca currents in the latter case (Angstadt and Calabrese, 1991; De Schutter et al., 1993). For spike-mediated transmission (I_{syns}), a double-exponential equation (equation 6.4) fitted to unitary IPSCs was used to describe the postsynaptic conductance associated with each action potential. The fraction of the postsynaptic maximal conductance reached after each spike was made a function of the past membrane potential to reflect the experimental observation that spike-mediated transmission varies in amplitude throughout a burst according to the baseline level of depolarization (Ivanov and Calabrese, 2000; A. A. Hill et al., 2001; Ivanov and Calabrese, 2003). Graded synaptic transmission (I_{synG}) was represented by a synaptic transfer function, which related postsynaptic conductance (the result of transmitter release) to presynaptic Ca^{2+} buildup and decline, via low-threshold Ca currents and a Ca^{2+} removal mechanism, respectively. For the intrinsic membrane and synaptic current equations used in our current implementation of this model, see A. A. Hill et al. (2001). Each cell was modeled as a single isopotential compartment with a leak conductance and a membrane capacitance.

The neuron model was at first hand tuned (only maximal conductances were considered free parameters) so that it produced tonic activity at a specified frequency in the absence of synaptic input but produced alternating bursting when configured

in a half-center oscillator circuit. Spike shape was not considered as important because somatic measurements of spikes do not reflect the full amplitude at the spike initiation sites in the majority of leech neurons and the model was single compartment. This state conformed to the experimental observation that heart interneurons recorded with sharp microelectrodes fired tonically when their inhibitory synaptic input was blocked with bicuculline (Schmidt and Calabrese, 1992) and much model development has proceeded with the neuron models so tuned.

Subsequent research using extracellular recordings showed that oscillator heart interneurons synaptically isolated in bicuculline can burst endogenously (Cymbalyuk et al., 2002). This observation called into question the leak parameters of the model neurons, and a careful mapping of the leak reversal potential (E_{leak})—not normally considered a free parameter—versus leak conductance (g_{leak}) for the neuron model revealed a narrow range of these conductances that could support endogenous bursting and a yet smaller region that produced bursting with the experimentally observed characteristics (Cymbalyuk et al., 2002). This analysis suggested that a small amount (1 nS or less) of nonspecific leak introduced by sharp microelectrode penetration was sufficient to suppress endogenous bursting. This example illustrates that every experimental constraint placed on a model will refine what is considered the canonical parameter set. Moreover, because endogenous bursting is not robust in many preparations, it is not clear that the canonical parameter set chosen is more than a snapshot of a particular preparation at a particular time; whether that snapshot is useful in providing mechanistic insight and experimental guidance depends on the flexibility of the canonical model (parameter set) in capturing the range of network output with physiologically relevant parameter changes.

Two-Cell Oscillator Model: The Half-Center Oscillator

Our approach in modeling the heartbeat timing network was to start with a (two-cell) half-center oscillator before building a more complex network model (figures 12.1c and 12.2) (A. A. Hill et al., 2002). This decision was made, not only for ease of implementation, but because we wished to explore the potential of the half-center oscillator to provide insights into how oscillations are generated by mutually inhibitory neurons, and we were convinced we would need these insights to guide our interpretations of activity within the timing network.

The free parameters in the model were the maximal conductance (\bar{g}) for each current (voltage gated or synaptic). The \bar{g} s were adjusted to be close to the average observed experimentally. The reversal potential, E , for each current was determined experimentally and was considered fixed (but see the earlier discussion regarding E_{leak}). The parameters of a canonical model were chosen so that the model's activity closely approximated the activity of the living neurons during normal oscillations (membrane potential excursions, period, intraburst spike frequency, and duty cycle),

in the passive response to hyperpolarizing current pulses, and in the reaction to current perturbations.

We then explored the dependence of model output on each of the \bar{g} s, which were varied bilaterally. We chose cycle period and average intraburst spike frequency of the oscillation as output measures for this analysis. These parameter variations, coupled with analysis of current flows during this activity, indicated that I_h and I_{Cas} were the main determinants of oscillator period (figure 12.1c). These observations led to specific predictions that were then tested and confirmed in the dynamic-clamp hybrid system experiments described in section 12.4. In this analysis it was instructive to vary parameters unilaterally in both the hybrid and the model half-center oscillators (Sorensen et al., 2004; Olypher et al., 2006). Parameter sweeps indicated that I_h is a likely candidate for neuromodulation controlling period in this system because period varies inversely and smoothly with \bar{g}_h (\bar{g}_h has only a small effect on spike frequency), and they identified \bar{g}_h as a parameter for variation in the timing network model to create a difference in period between the two component half-center oscillators (see later discussion). Moreover, recent experiments show that the peptide myomodulin, which speeds up the heartbeat timing network, acts in part by increasing \bar{g}_h (Tobin and Calabrese, 2005).

Building the Timing Network

We constructed a model of the beat timing network based on our half-center oscillator model that corresponds to the mutually inhibitory pairs of interneurons in the third and fourth ganglia (figure 12.2). The coordinating interneurons of the first and second ganglia that couple the segmental oscillators were modeled, like the oscillator interneurons, as single compartments. Although their firing properties have been quantitatively described (Masino and Calabrese, 2002b,c), no voltage-clamp data are available for the coordinating interneurons. Therefore, these neurons were modeled using current equations from the oscillator heart interneuron model, so that they exhibit spiking behavior similar to that of living neurons. They contain only spiking currents, and leak, and receive and send only spike-mediated synaptic inhibition. The coordinating neurons are capable of initiating spikes at sites in both G3 and G4. In the first-generation timing network model, we did not include this detail and have considered the coordinating fibers to be single compartments (G4 sites) that receive inhibition from ipsilateral G3 and G4 interneurons (figure 12.2, simple CFs). This decision was made to understand better the basic operation of the system, with the plan that extra detail would be added if needed to capture network dynamics.

This model was developed to assist our experimental analysis of how the intersegmental phase lag seen in figure 12.1b is generated. To explore the generation of this phase difference, we first created two “segmental” oscillators that correspond to G3 and G4 half-center oscillators and their coordinating fibers within one ganglion; i.e.,

two six-cell models (figure 12.2) (A. A. Hill et al., 2002). The (six-cell) segmental oscillators behaved similarly to the (two-cell) half-center oscillators except the period was longer (~25%); a parameter variation of the \bar{g}_{synS} for the synapse between coordinating interneurons and oscillator interneurons indicated that this slowing is due to “extra” inhibition in the half-center oscillator. Next, two such segmental oscillators were combined to form a single eight-cell oscillator representing the complete beat timing network (figure 12.2). An advantage of this approach is that it is possible to determine the inherent cycle periods of the segmental oscillators in isolation before coupling them to determine the ensemble behavior of the eight-cell network.

Experimental analysis suggested that the phase lag was associated with differences in the inherent cycle periods of the segmental oscillators (Masino and Calabrese, 2002b,c). In the model, the inherent cycle periods of the segmental oscillators were varied by altering \bar{g}_h bilaterally in model G4 oscillator neurons; a decrease or increase in \bar{g}_h from the canonical level led to a lengthening or shortening of the period of the G4 segmental oscillator (A. A. Hill et al., 2001). When two segmental oscillators with different inherent periods were combined, the cycle period of the eight-cell timing network was very close to that of the faster oscillator’s period. The faster oscillator consistently led the slower one, and the phase lag between the oscillators was proportional to the difference in period between the two segmental oscillators (A. A. Hill et al., 2002). These observations in the model then became experimental predictions or hypotheses.

These hypotheses were tested in the living system in which the segmental oscillators were uncoupled and recoupled by reversible blockade of action potential conduction between the ganglia and were confirmed (Masino and Calabrese, 2002c). One other model prediction was that entrainment of one segmental oscillator by driving the other with rhythmic current pulses could only be to a shorter period (A. A. Hill et al., 2002). Experimental analysis showed that entrainment could be effected to both shorter and longer periods, and moreover indicated that the period range of entrainment was greater for driving the G3 oscillator than for driving the G4 oscillator (Masino and Calabrese, 2002a). These findings indicated that the model was not adequate for capturing true network activity under experimental test.

We responded to this “model failure” by increasing the sophistication of our model of the coordinating interneurons (figure 12.2, extended CFs) (Jezzini et al., 2004). We now modeled them as intersegmental axonal processes with separate spike initiation and synaptic input sites in G3 and G4. In accord with experimental data, the G3 site had a slightly higher inherent spike frequency and received input only from the G3 oscillator interneuron, while the G4 site had slightly lower inherent spike frequency and received input from both the G3 and G4 interneurons (Masino and Calabrese, 2002b). These improvements brought the model in alignment with our entrainment analysis (figure 12.1d).

Conclusions

In the approach to modeling the timing network for the leech heartbeat, there was a close interplay of modeling and experiment, so that insights derived from experiments were used to refine the model and model-generated hypotheses led directly to experiments. A network model was built up in a step-by-step fashion, adding more circuit and cellular detail as it was needed and made available from experiments. In our case, these steps reflected our conceptions of how the living system was organized—oscillator and coordinating heart interneurons, half-center oscillators, segmental oscillators, and timing network—and the successes of our models corroborated these conceptions. Each step in the model-building process was used to gain mechanistic insight through analysis of the model, and in this analysis parameter variation became a tool for probing network function. Our conceptions about how the network worked focused attention on key parameters; these were then systematically varied in the model to generate testable predictions or hypotheses about the living system.

12.3 Parameter Nonuniqueness

The preceding sections described how to construct a canonical neuronal network model that reproduces a desired activity pattern, often derived from experimental observation. However, the electrical activity patterns generated by the same neuronal network in different animals typically show variability. For example, a survey of the rhythmic motor patterns generated by the pyloric circuits in stomatogastric ganglia from ninety-nine lobsters shows that pyloric periods and other motor pattern characteristics, such as burst durations, spike frequency within the burst, number of spikes per burst, and phase relationships, can vary severalfold among animals (Bucher et al., 2005). Modeling studies at the cellular (Bhalla and Bower, 1993; Foster et al., 1993; Goldman et al., 2001; Prinz et al., 2003; Achard and De Schutter, 2006) and network levels (Prinz et al., 2004a) have shown that electrical activity in a model within the biological range can arise from widely varying sets of neuron or network parameters, indicating that neuronal systems can use a variety of “solutions” to achieve functional behavior, rather than having to be fine tuned to a unique set of parameters.

The notion that networks produce functional activity in different animals—or in the same animal at different times—on the basis of different parameter sets is supported by experimental evidence from the crustacean stomatogastric ganglion, where voltage-clamp experiments (Golowasch et al., 1999, 2002) and measurements of ion channel mRNA expression levels (Schulz et al., 2006) in the same neurons identified in different animals have revealed that specific ionic membrane conductances and their mRNA levels can vary severalfold among animals while still producing similar

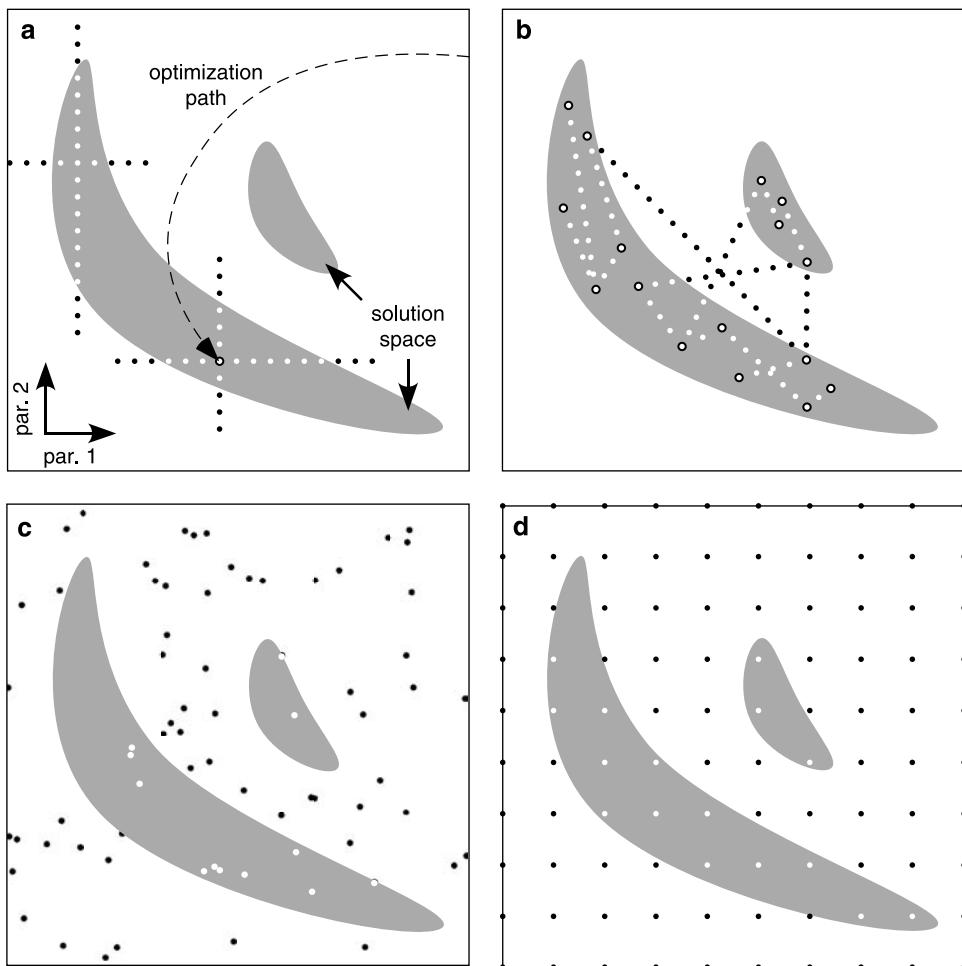
network outputs. This notion is also supported by current amplitude measurements in dissociated cerebellar Purkinje neurons, which show variable current amplitudes even within groups of neurons with very similar burst properties (Swensen and Bean, 2005).

Mapping Solution Spaces

If multiple parameter sets can produce similar, functional network behavior, how can we identify the complete “solution space” of a neuronal system? Several approaches for the mapping and exploration of solution spaces have been developed in the past, but most of them have been tested at the level of single neuron models and have not yet been applied at the level of small networks. However, the challenges of exploring the solution space of a small network model seem quite similar to the challenges of exploring the solution space of a complex single-neuron model. The methods discussed here, even those not yet applied to small networks, should therefore be easily generalized to the network level.

A frequently used method (Bhalla and Bower, 1993; Cymbalyuk et al., 2002; Achard and De Schutter, 2006) assesses how sensitively model behavior depends on the model’s parameters by starting with a canonical parameter set—obtained with one of the parameter search methods described in chapter 2—and varying one parameter at a time while keeping all others fixed. As figure 12.3a illustrates, this method has several potential pitfalls. If the true solution space consists of several disconnected parameter regions or has a concave shape in a high-dimensional parameter space, variation of one parameter at a time may miss portions of the solution space. Furthermore, conclusions about the sensitivity of model behavior to variations in the underlying parameters cannot necessarily be generalized from one region of solution space to other regions. This is illustrated by the parameter variations represented by the dots in figure 12.3a. Variation of one parameter at a time around the optimized model at the tip of the arrow will indicate that the model’s behavior depends more sensitively on parameter 2 (vertical direction in the schematic) than on parameter 1 (horizontal direction) because the range of values that generate functional behavior is narrower for parameter 2 than for parameter 1. While this is a valid conclusion in the vicinity of the optimized model, the opposite is true in other regions of parameter space, such as the top left part, where the model’s behavior depends more sensitively on parameter 1 than on parameter 2.

Another approach is to randomly vary the parameters of a neuronal model system and thus perform a stochastic search of the model’s parameter space to identify parameter combinations that yield behavior in the desired range (Foster et al., 1993). This approach is illustrated in figure 12.3c in two dimensions and described in detail in chapter 2, section 4. Alternatively, the parameter space can be covered with a grid of simulations (figure 12.3d and chapter 2, section 3), again looking for grid points

**Figure 12.3**

Mapping solution space. White boxes: schematic representation of parameter space. Gray areas: solution regions. Dots: simulated parameter sets identified as solutions (white) or nonsolutions (black) or found as solutions through optimization methods (white center, black surround). (a) How sensitively model behavior depends on model parameters can be explored by varying one parameter at a time, starting from a known solution; for example, one identified by an optimization method. (b) After running evolutionary algorithms several times with different results, the solution space can be mapped by exploring hyperplanes spanned by subsets of results (dotted lines in this two-dimensional schematic). (c) Stochastic parameter space exploration. (d) Systematic parameter space exploration on a grid.

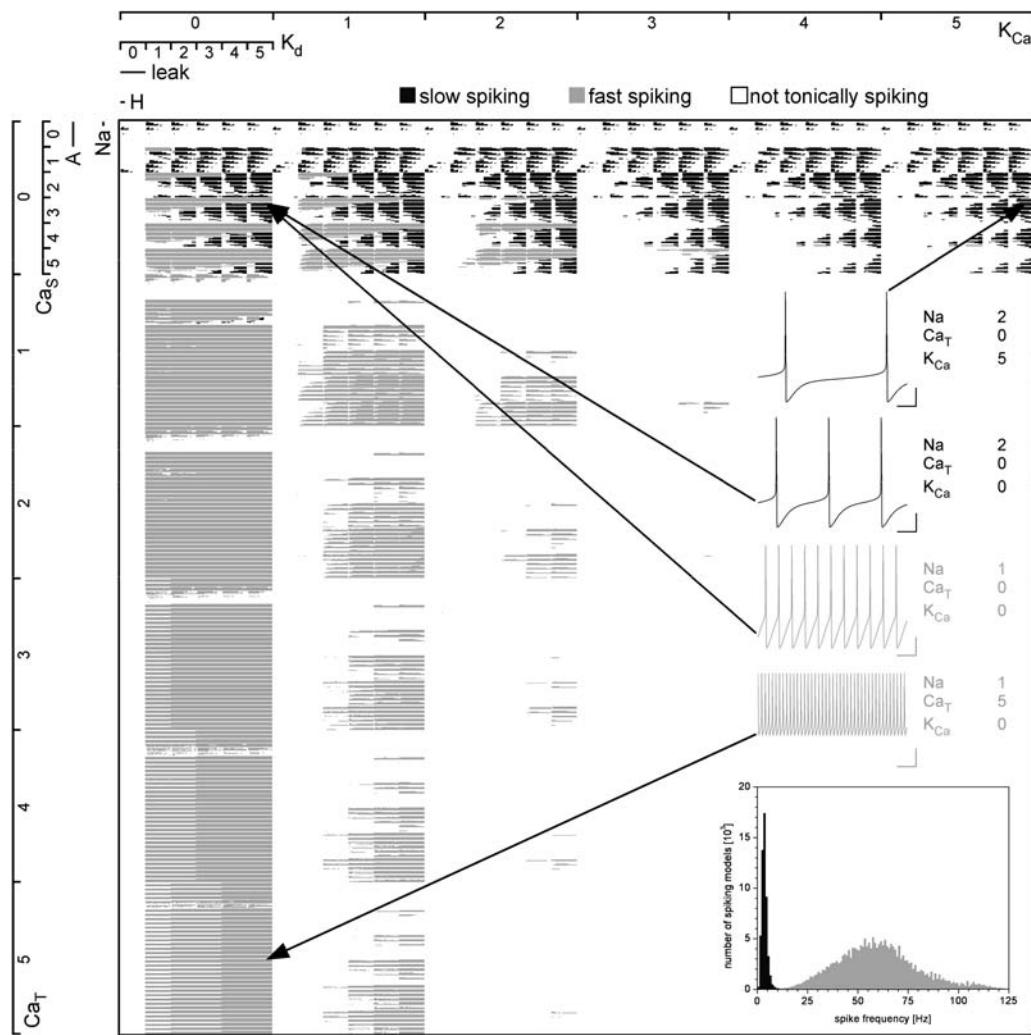
that generate the target activity (Goldman et al., 2001; Prinz et al., 2003, 2004a). Depending on the number of model parameters involved, both approaches require large numbers of simulations and have thus only recently become feasible for high-dimensional models such as network models with multiple component neurons (Prinz et al., 2004a). However, a comparison of c and d in figure 12.3, which contain the same number of sampling points, suggests that a grid may be more reliable at approximating the true shape of solution space because it has a fixed spacing of sampling points and thus is guaranteed not to miss large regions of parameter space (as happened at the top left of the gray solution space in figure 12.3c).

Brute-force computational exploration methods require rapidly increasing numbers of simulations as the dimensionality of the model neuron or network increases. Furthermore, the solution spaces of complex models can be of complex shape, so that exploring parameter space with sparse, randomly distributed sampling points or with samples on a sparse grid can miss salient features of the solution space. This was recently pointed out in a study that applied an evolution strategy to a complex model of cerebellar Purkinje cells (Achard and De Schutter, 2006, 2008). The population of twenty solutions resulting from multiple runs of the evolutionary algorithm subsequently served as anchor points for hyperplanes (symbolized by lines in the two-dimensional cartoon in figure 12.3b) that were explored at high spatial resolution through additional simulations. This elegant hybrid approach (see chapter 2, section 5 for the concept of hybrid parameter search methods) led to a detailed picture of the complex solution space of the model. The authors argue that for highly complex models with many varied parameters, such complex solution spaces may be difficult to identify with the coarse-grained stochastic and grid methods described here.

Visualizing Solution Spaces and Other High-Dimensional Datasets

Both stochastic and systematic explorations of neuron and network parameter spaces result in lists of parameter combinations that generate similar behavior. Understanding where these solutions are located in parameter space is challenging, owing to the high-dimensional nature of the parameter spaces of all but the simplest neuronal system models. Analyses of solution spaces therefore often make use of lower-dimensional projections of parameter space or slices through it, which carries the risk of missing the larger picture of the solution space topology. However, recent advances in visualization tools for high-dimensional datasets are beginning to provide alternatives that are better suited for capturing the complexity of high-dimensional parameter spaces.

A technique called dimensional stacking (LeBlanc et al., 1990) was recently applied to a neuronal model database to determine how neuronal models with certain types of behavior are arranged in parameter space (Taylor et al., 2006). Figure 12.4 is an example of a dimensional stack that shows two solution spaces, for fast and

**Figure 12.4**

Visualizing solution spaces. This dimensional stack of an eight-dimensional neuronal conductance space shows the location of the solution spaces for slow (<11.5 Hz) and fast (>11.5 Hz) tonic spiking. Each model in the database of 1.7 million entries is represented by a black pixel if the model is a slow spiker, a gray pixel if it is a fast spiker, and a white pixel for all other behaviors. The pixels are arranged in two dimensions as described in the text. The inset shows a spike frequency histogram of the two spiker categories. (Dimensional stack courtesy of Adam L. Taylor.)

slow tonic spiking of a model neuron. The stack shows one pixel for every parameter set in a database of different versions of the model neuron that was generated by systematically exploring the eight-dimensional space of maximal membrane conductances of the model neuron on a grid, as in figure 12.3d (Prinz et al., 2003). Each pixel is either black (for low-frequency tonic spikers), gray (for fast tonic spikers), or white (for all other activity types). At the highest level of organization, the pixels are arranged in six by six large blocks, with the I_{KCa} maximal conductance \bar{g}_{KCa} and the I_{CaT} maximal conductance \bar{g}_{CaT} constant in each block and \bar{g}_{KCa} increasing from block to block from left to right and \bar{g}_{CaT} increasing from top to bottom, as indicated by the scales at the top and left of the Figure. Within each of these large blocks, the pixels are arranged into six by six smaller blocks according to their levels of \bar{g}_{Kd} (left to right) and \bar{g}_{Cas} (top to bottom), and so on through two more levels of organization for the remaining conductance parameters. The dimensional stack thus shows all entries in the eight-dimensional model database on a two-dimensional plot without projecting, averaging over dimensions, or slicing through parameter space.

Dimensional stacking visualizations of model datasets reveal information about the location of solution spaces in parameter space that would not otherwise be obvious. In figure 12.4, the location of the black pixels almost exclusively in the top sixth of the stack indicates that slow spiking requires that I_{CaT} , a fast transient Ca^{2+} current, be mostly or completely absent from the membrane. Furthermore, the absence of black pixels from the lower left corners of the blocks of constant \bar{g}_{CaT} and \bar{g}_{KCa} suggests that slow spiking is not possible if \bar{g}_{Cas} (a slower Ca^{2+} conductance) is high and \bar{g}_{Kd} is low at the same time. The vertical white stripe at the left edge of the stack and similar white stripes for higher values of \bar{g}_{KCa} indicate that fast spiking is not possible without I_{Kd} . Finally, the absence of gray pixels from the horizontal stripe for zero \bar{g}_{CaT} and zero \bar{g}_{Cas} at the top of the stack shows that nonzero total Ca^{2+} conductance is necessary for fast spiking. Additional conclusions can be drawn from closer inspection of figure 12.4 at lower levels of organization or from visualization of the same data in a dimensional stack with a different stacking order, i.e., with different assignment of the conductance parameters to the different levels of organization in the stack (Taylor et al., 2006).

Analyzing Connectedness of Solution Spaces

Dimensional stacking displays high-dimensional datasets in two dimensions, but it is not informative about whether a solution space is connected or consists of multiple disconnected regions in parameter space. The question of connectedness is important from a biological perspective because neural systems that need to continuously regulate their parameters to achieve a desired behavior during development or maintain it in the adult can do so with simpler regulation strategies if their solution space is continuous rather than scattered. Similarly, modeling a neuron or small network and

exploring its solution space is easier if that space is contiguous (see, for example, figure 12.3a).

For model databases derived from exploration of parameter space on a grid, connectedness can be assessed with a connected-components algorithm from the image-processing literature (Jain et al., 1995). Applied to a grid-based model dataset, this algorithm considers two grid points as part of the same connected component of solution space if they can be connected by a path on the sampling grid so that each grid point along the path is also a solution (Taylor et al., 2006). Fortunately, most existing mappings of solution spaces in neuron and network models seem to indicate that parameter combinations that generate similar functional behavior tend to be localized to a single connected component (Prinz et al., 2003; Taylor et al., 2006). If, in contrast, several disconnected components are found, it may be worthwhile to refine the sampling grid to examine whether this disconnectedness is an artifact of coarse sampling (Prinz et al., 2003; Taylor et al., 2006).

It should be noted that even if a solution space is fully connected, it can have very complex topology. This was recently demonstrated for the solution space of a complex, multicompartment Purkinje cell model whose solutions were required to replicate a narrowly constrained set of electrical behaviors (Achard and De Schutter, 2006). In this study, the solution space was found to consist of loosely connected hyperplanes, which raises the question of how a homeostatic regulatory mechanism would have to be structured to support robustness of a complex behavior in a complex neuron.

12.4 Testing Network Model Predictions with a Dynamic Clamp

Network models are most valuable if they are grounded in physiological reality through repeated cycles of model refinement and testing of predictions in the living network. The dynamic clamp (Sharp et al., 1993; Prinz et al., 2004b) is a useful tool that enables such tests by allowing the researcher to combine model and living components in hybrid networks. A dynamic-clamp system is a real-time connection between a living neural system and a computational model in which the membrane potential recorded from one or several living neurons is continuously fed into the computer; the differential equations describing the model are numerically solved; and the resulting membrane or synaptic currents (e.g., equations 5.6 and 6.10) are fed back into the living neuron(s) at temporal resolutions of several to tens of kilohertz (figure 12.5a). Note that a dynamic clamp is based on a conductance model of currents, which differentiates it from standard current injection (chapter 6, section 1).

Different hardware and software implementations of the dynamic clamp are available and their advantages and disadvantages have recently been compared for specific dynamic-clamp applications (Prinz et al., 2004b). Examples of the available systems are listed in table 12.1. System features to be considered when choosing a

Table 12.1
Examples of dynamic-clamp implementations

		Windows-based	Real-time Linux-based ^a	Embedded processor or DSP	Devices
References	Rabbah and Nadim (2005)	Nowotny et al. (2006)	Dorval et al. (2001)	Raikov et al. (2004)	Kullmann et al. (2002)
URL	http://stg.rutgers.edu/software/index.html	http://inls.ucsd.edu/~nowotny/dynclamp.htm	http://www.bu.edu/ndl/dynamicclamp.html	http://www.neuro.gatech.edu/mrci/	NA
Programming language	LabWindows/CVI	C++	C, C++ for user interface	MRCI modeling language, C	Real-time labview
Update rate ^b	5 kHz	10 kHz	20 kHz	30 kHz	40 kHz ^c
Existing applications	Artificial modulated conductances; artificial chemical or electrical synapses between up to two cells and one model neuron; hybrid two-cell or three-cell networks	Artificial conductances; artificial learning chemical or electrical synapses between up to four cells; waveform generation; scripting	Artificial conductances; hybrid two-cell networks; adding multiple compartments	Artificial synaptic inputs; hybrid two-cell networks	Artificial conductances; artificial chemical synapses; recording current-voltage curves
Number of channels ^d	Two in, two out	Four in, four out	Two in, two out	Two in, two out	Four in, four out
User interface	Graphical	Graphical	Graphical	Command line	Graphical
Saves traces?	Yes	No	Yes	Yes	NA
Displays traces?	Yes	Yes ^e	Yes	No	Yes

Source: Updated and modified from (Prinz et al., 2004b), with permission.

Abbreviations: DSP, digital signal processing; MRCl, model reference current injection; NA, not available.

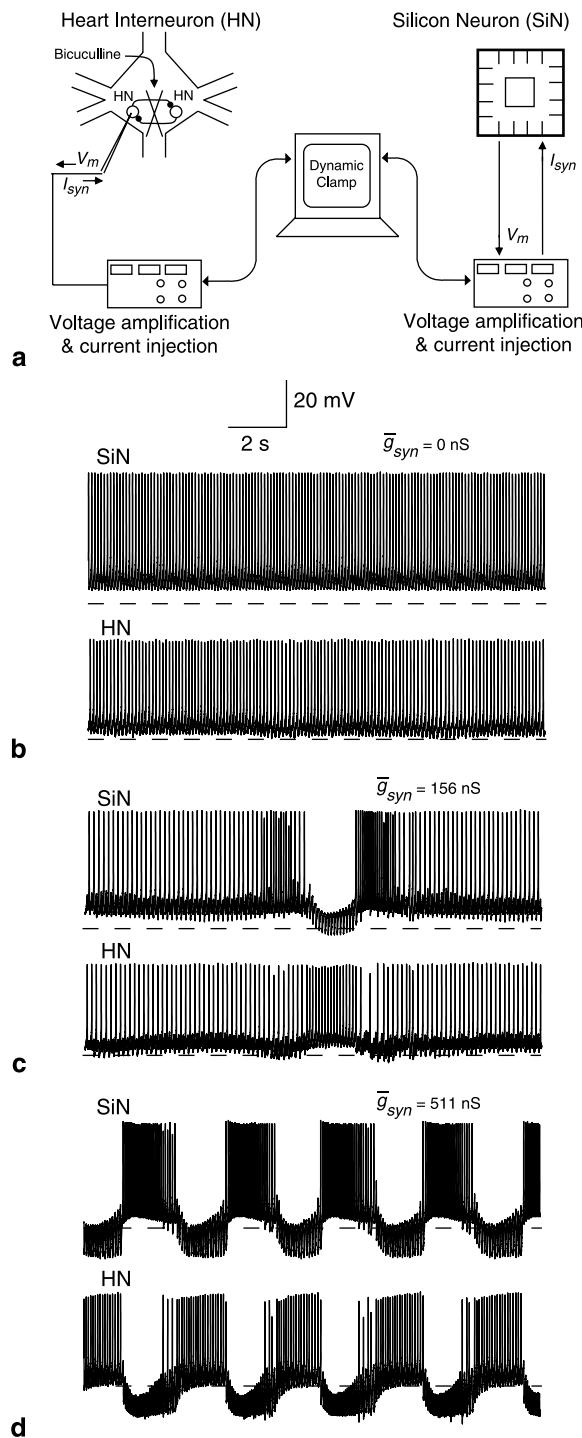
^aA merged version of the two real-time Linux-based systems described here is currently under development and is documented at <http://www.ritxi.org/> (R. Butera and J. White, personal communication).

^bUpdate rates vary, depending on the computational load. Update rates given here are maximum values of published versions of the system and will increase with time.

^cNewer, unpublished versions of this system can achieve update rates of up to 40 kHz, and can handle as many as sixteen input and two output channels (J. White, personal communication).

^dChannel numbers given here are those of published versions of the systems. Most systems can be modified to handle larger channel numbers if different hardware is used.

^eDuring dynamic-clamp operation, the use of the display function is discouraged because it can affect system performance.



dynamic-clamp implementation for a particular application include speed, number of input and output channels (which determines the number of living cells that can be dynamic clamped simultaneously), ease of use, and cost. The rate at which a dynamic-clamp system reads the membrane potential(s) of the connected neuron(s), performs the necessary calculations, and feeds the appropriate current(s) back into the biological cell(s) can be a crucial consideration when fast neuronal events, such as the gating of fast sodium currents, are to be simulated. Analog devices such as the circuit described in Chance et al. (2002), which perform the necessary calculations in dedicated electrical circuits rather than digitally, are unsurpassed when it comes to dynamic-clamp speed, but usually have the disadvantage of being hard to adapt to new applications. In contrast, real-time Linux-based systems combine relatively high update rates and flexibility, but often require significant computer expertise from the user. Similarly, embedded processors of digital signal processors (DSPs) achieve high update rates and are highly flexible, but often require substantial financial investments. Finally, Windows-based dynamic-clamp systems usually operate at intermediate speed and are mostly user friendly and affordable, but have the downside of being subject to occasional operating system interrupts that can cause disruptions in a smooth dynamic-clamp operation.

Depending on the nature of the computational model in the computer, the dynamic clamp can be used in a variety of configurations. These range from the addition of a virtual membrane or synaptic conductance to a real neuron (Cymbalyuk et al., 2002), through the coupling of several neurons with artificial synapses (see figure 12.5), to the incorporation of living neurons into networks of model neurons (LeMasson et al., 1995).

Testing the Role of Intrinsic Properties in Network Behavior

The dynamic clamp can be used to selectively increase or decrease specific membrane conductances in neurons embedded in a network. It can therefore be used to determine how a particular membrane conductance in a network neuron shapes

Figure 12.5

Dynamic-clamp synapses in a hybrid network. (a) Experimental configuration. A leech heart (HN) interneuron is synaptically isolated from its contralateral partner by application of bicuculline, impaled with a microelectrode, and connected to a dynamic-clamp computer via an amplifier. Simultaneously, a silicon neuron is also connected to the dynamic clamp via an amplifier. The dynamic-clamp computer contains a model of mutual inhibitory synapses between the HN neuron and the SiN; it computes the synaptic currents in real time, depending on the momentary membrane potentials of the HN neuron and SiN, and injects those currents into the HN neuron and SiN via the amplifiers to create artificial inhibitory synapses between them. (b) Potential traces of HN neuron and SiN when not connected ($g_{syn} = 0$ nS) via a dynamic clamp. Both spike tonically. (c) Weak reciprocal inhibition ($g_{syn} = 156$ nS) leads to sporadic lapses in activity. (d) Strong mutual inhibition ($g_{syn} = 511$ nS) results in stable alternating bursting similar to the activity of the living neuron pair in the absence of the synapse blocker. Dashed lines in b–d indicate -50 mV. (Modified from Sorensen et al., 2004, with permission.)

the network's behavior. A study in the pyloric CPG kernel of lobsters used this technique to verify that the changes in network activity after overexpression of the hyperpolarization-activated cation current I_h are consistent with the effects of increased I_h maximal conductance, both in a hybrid network—by adding artificial I_h to the living pacemaker neurons—and in a computational model of the pacemaker kernel (Y. Zhang et al., 2003). It may appear redundant to determine the impact of a membrane current both in a computational network model and by dynamic-clamp addition of the current to the living network. However, the effect of a current on network behavior depends on its interactions with the other currents present in the system. Because any computational network model is by necessity a simplification of the living network, dynamic-clamp verification of modeling results in the living network is advisable and makes those results more reliable.

In contrast to pharmacological blockers of specific membrane currents, which mainly affect the amplitude of a current, the dynamic clamp allows manipulation of all other voltage-dependent and dynamic properties of an artificially added conductance in a biological context. Sorensen et al. (2004) used this approach in a hybrid network of a living and a model leech heart interneuron to show that not only the amplitude but also the activation and deactivation time constant of I_h determines the heartbeat period and the balance of activity between the left and right sides of the heartbeat network. This finding corroborated the results of a purely computational model of coupled heart interneurons (A. A. Hill et al., 2001). Also in the leech heartbeat system, the dynamic clamp was used in a hybrid network configuration to show that the inactivation time constant of the low-voltage-activated calcium current I_{CaS} controls burst duration both in the model and in the living part of the network (Olypher et al., 2006).

In cases where no specific blockers for a membrane current are available, cancellation of that current in the living system with the dynamic clamp can be the only possibility for testing the predictions of a model. A model of a CA1 hippocampal pyramidal neuron predicts that the persistent sodium current I_{NaP} amplifies the after-hyperpolarization following a spike, reduces the gain of the frequency-current relationship in response to steady current injections, increases the regularity of repetitive firing, and reduces spike timing precision in response to excitatory inputs from the surrounding network (Vervaeke et al., 2006). Dynamic-clamp subtraction of I_{NaP} from living CA1 pyramidal neurons corroborated these predictions and thus verified that the model accurately depicts the behavior of CA1 pyramidal neurons in a network. When using the dynamic clamp to subtract a voltage-dependent conductance from a living neuron in this manner, special care must be taken to ensure that the voltage dependence and temporal dynamics of the model's current to be subtracted closely match those of the current in the living neuron because any discrepancies can lead to a net current remaining uncanceled.

Exploring Synaptic Contributions to Network Dynamics

Network activity is shaped by the intrinsic properties of the component neurons and by the properties of the synaptic connections between them. Again, the dynamic clamp is a valuable tool that allows the controlled modification of synaptic inputs to a living neuron to be used to assess how synaptic properties influence network output (see figure 12.5).

When testing the effect of excitatory and inhibitory inputs on living neurons, it is important to take into account that realistic synaptic inputs are conductance events, not current pulses (chapter 6, section 1). For example, reaching a certain level of depolarization in the living neuron requires more excitatory conductance inputs than current inputs because any depolarization reduces the driving force for further excitatory conductance inputs. Synaptic signals arriving at a living neuron should therefore always be simulated as conductances, not currents (Dorval and White, 2006).

Depressing synapses can mediate bistability in oscillatory networks with recurrent inhibitory synapses, thus endowing the network with a larger repertoire of activity patterns. This was demonstrated in a computational model of two reciprocally inhibitory model neurons and verified in hybrid networks of a living pacemaker neuron and a model neuron connected by depressing inhibitory synapses via a dynamic clamp (Manor and Nadim, 2001). Because the voltage range covered by a neuron can vary from cell to cell, the activation range of artificial synapses from a living neuron has to be manually adjusted for each dynamic-clamp experiment involving synapses.

In cortical networks, where individual pyramidal neurons receive synaptic inputs from many other neurons, balanced excitatory and inhibitory inputs can modify how a neuron respond to inputs by modulating the gain of its frequency-current relationship. Using a dynamic clamp to mimic the barrages of excitatory and inhibitory inputs received by pyramidal neurons *in vivo*, Chance et al. (2002) demonstrated this gain modulation *in vitro*, which is consistent with an analytical model of gain modulation in an integrate-and-fire neuron. Dynamic-clamp studies have furthermore shown that the statistics, dynamics, and balance of excitatory and inhibitory synaptic inputs to cortical neurons tune the variability and reproducibility of their responses (Badoual et al., 2005; Piwkowska et al., 2005; Dorval and White, 2006), their spike timing, and the frequency of network activity (Aradi and Maccaferri, 2004).

Studying Network Synchronization with a Dynamic Clamp

Because of their role in normal and diseased neuronal network function, synchronization phenomena and mechanisms are of considerable interest and have repeatedly been studied computationally and with the dynamic clamp (X.-J. Wang and Buzsaki, 1996; Lewis and Rinzel, 2003; Bem et al., 2005; Merriam et al., 2005; Netoff et al.,

2005; Preyer and Butera, 2005). Temporal delays and rise and fall time constants of synaptic inputs that are due to remote synapse locations can be important factors for network oscillations. When studying oscillation mechanisms with a dynamic clamp, the impact of delays and variations in time constants should therefore be explored (Netoff et al., 2005). Two limitations on the use of a dynamic clamp in tests of network model predictions are that simultaneous recordings become technically difficult if more than a few neurons are involved and most dynamic-clamp systems are limited to two output channels (Prinz et al., 2004b).

Several recent studies have overcome these limitations by iteratively constructing semisynthetic neuronal networks that replace simultaneous dynamic clamping of multiple neurons by consecutive dynamic clamping of a single living neuron, using the neuron's outputs from earlier sweeps to construct its inputs in later sweeps (Reyes, 2003). A comparison of the network activity in semisynthetic circuits constructed in this manner with the activity in model networks has demonstrated the validity of this approach and has helped us to explore how cellular, network, and synaptic factors contribute to synchronization in networks of inhibitory neurons (Sohal and Huguenard, 2005).

A disadvantage of semisynthetic circuits constructed from a single living neuron is that they contain, by definition, homogeneous cell populations. Heterogeneous cellular and synaptic properties have been determined to be important factors in sub- and suprathreshold network oscillations (Golomb and Rinzel, 1993). In the inferior olive, a model network that electrically couples neurons with heterogeneous ion channel densities predicts that the oscillations of the subthreshold membrane potential observed in the inferior olive arise from interactions of heterogeneous neurons and do not require the presence of intrinsically oscillating neurons (Manor et al., 1997), a prediction that has since been corroborated by dynamic-clamp experiments (Manor et al., 2000). In the hippocampus, the presence of heterogeneous subpopulations of inhibitory interneurons has been shown to influence synchronization of pyramidal cells in computational models (Aradi and Soltesz, 2002), a prediction that again has been verified with a dynamic-clamp approach (Foldy et al., 2004).

12.5 Mechanosensory Feedback

Neuronal networks are in most cases studied in isolation from sensory inputs and detached from the muscles that translate motor neuron activity into movement in the intact animal. Consequently, most network models focus on the interactions of nerve cells and synapses, but do not include the sensory or motor periphery. Although many neuronal networks produce similar activity in the presence or absence of sensory and motor components, it is becoming increasingly obvious that in some cases the behavior of a neuronal network can be significantly influenced by the physical

properties of the body in which it is embedded (Chiel and Beer, 1997). This has especially been acknowledged in the case of circuits involved in the generation of motor patterns (Holmes et al., 2006). Focusing on motor systems, we describe here how the inclusion of the motor periphery and sensory feedback in network models can validate circuit models and lead to new insights into mechanisms of motor pattern generation.

Open-Loop Neuromechanical Models

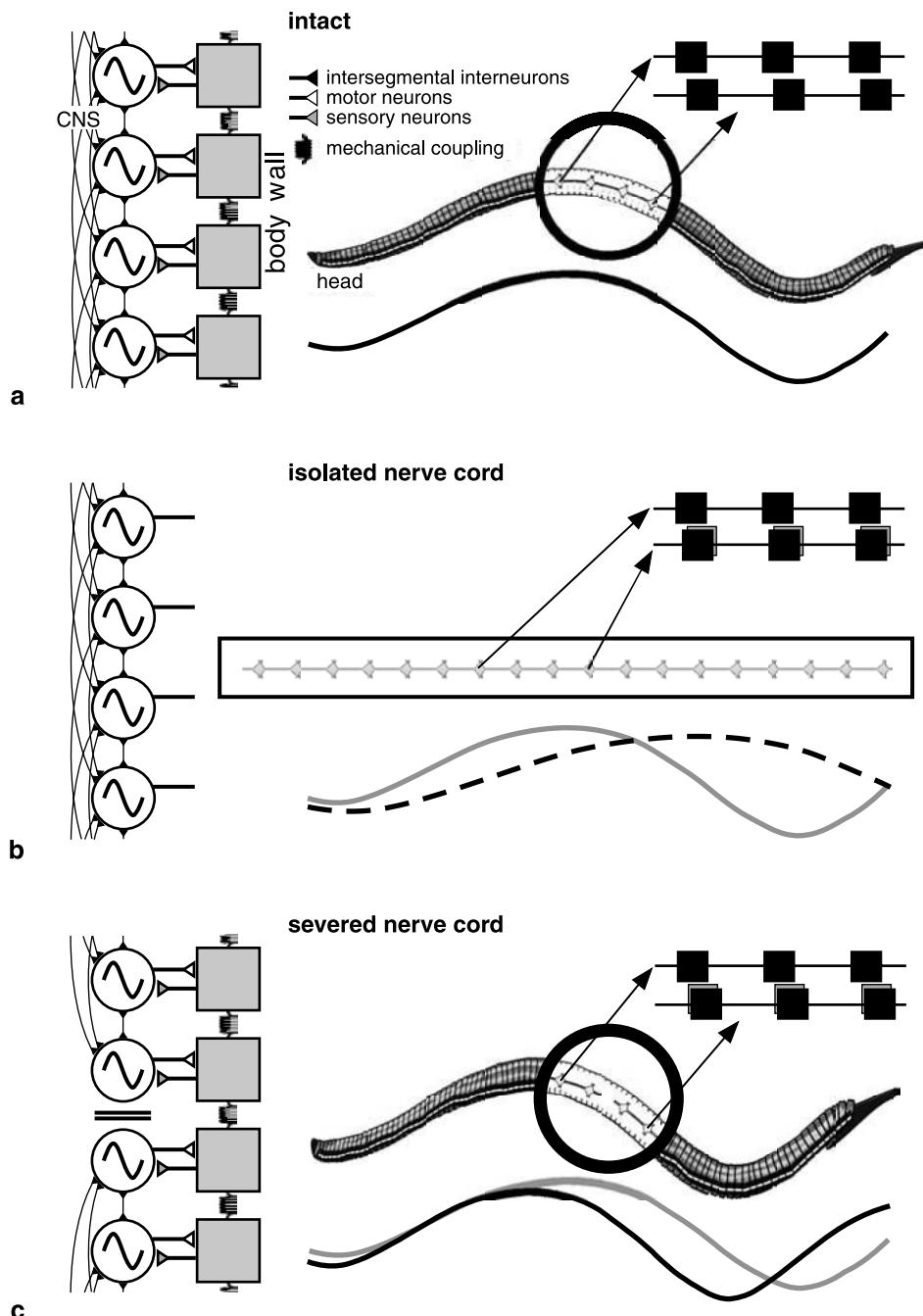
The simplest network model configuration that includes the motor periphery is an open-loop system in which the neuronal circuit component of the model generates electrical activity that drives movements of the model's motor component without receiving feedback from the periphery. Such unidirectional models can be used to validate the circuit model by examining the movements it generates when controlling a virtual "body" and comparing them with the movements of the animal to be modeled. This approach was recently used in a model of zebrafish locomotion that reproduced different behaviors, such as normal swimming and escape (Kuo and Eliasmith, 2005).

In the *Aplysia californica* feeding system, the feeding apparatus consists of a grasping structure—the radula—and the underlying muscle tissue and is controlled by a small neuronal network. Despite its relative simplicity, the neuronal feeding circuit can generate a range of movements of the feeding apparatus, including biting, swallowing, and rejection of food. Neuromechanical models of the feeding circuit and apparatus suggest that the neuronal commands generated by the feeding circuit are optimized to make use of mechanical hinge properties and muscle function (Sutton et al., 2004a,b). Furthermore, an open-loop model of the feeding system's neuromuscular transform has shown that complex neuromodulation of the motor periphery increases the number of possible movements and is necessary for functional network performance (Brezina et al., 2005).

What form the mechanical part of a neuromechanical model should take depends on the type of question to be addressed with the combined model. For example, models intended to examine the movement of a limb under the control of a neuronal network require a neuromuscular transform that translates neuronal activity into muscle contraction, and a mechanical model of the limb that is moved by the contracting muscle (Ekeberg, 2001). If the model is to describe the locomotion of an entire body, mechanical models of body parts, the environment, and muscle tension may have to be specified.

Closed-Loop Models

If the neural circuit controlling a movement receives proprioceptive or other sensory feedback from the body part it controls, or if there is mechanical feedback from the



environment, such as water forces during swimming, the motor system operates in a closed-loop configuration. Models of closed-loop sensorimotor systems range from the level of a single or a few body parts—limbs or organs—to computational models of the entire body in its environment that are concerned with the coordination among segments or limbs and mechanical interactions with surrounding surfaces, gases, or fluids. Ideally, models of sensory feedback to the neuronal component of the closed-loop system should incorporate the response properties of the living sensors to be modeled and reproduce the living system's projection patterns of sensory feedback onto the components of the neuronal circuit. In the absence of experimental data to guide construction of sensorimotor models, general design principles—such as mutual inhibition between antagonistic sensorimotor pathways or system stabilization through negative feedback—should be followed.

At the level of a single limb, a dynamic 3-D biomechanical model of a stick insect leg has been integrated with a model of the neural circuit controlling it and properly timed proprioceptive feedback signaling (Akay et al., 2004; Ekeberg et al., 2004). The integrated model accounted for the entire step cycle observed in the living system under different walking conditions. Time delays of sensory feedback signals in a sensorimotor loop have also been shown to have important influence on the mutual entrainment of CPG and limbs in a model of bipedal locomotion (Ohgane et al., 2004). Including mechanical models of the periphery and sensory feedback in models of motor systems can also reveal mechanisms that determine oscillation frequency. For example, closed-loop coupling of a CPG to a pendulum (as a proxy for a limb) with sensory feedback can shift the oscillation frequency of the combined system away from the intrinsic CPG frequency and in the direction of the pendulum's resonance frequency (Iwasaki and Zheng, 2006). Furthermore, sensory feedback from a mechanical system controlled by a neural system can lead to stable oscillations of the combined model even if the neural component is not in itself oscillatory (Sekerli and Butera, 2005).

Although most of the models described thus far address locomotion, other models of motor systems also benefit from the inclusion of feedback loops. One example is a

Figure 12.6

Role of sensory feedback in leech swimming. (a) Swimming in the intact leech. The swim muscles in each body segment are controlled by a neural oscillator; oscillators in different segments are connected via intersegmental interneurons. Each oscillator receives sensory feedback from the body wall of its segment, and the body walls in neighboring segments are mechanically coupled. The intact system generates rhythmic activity that is phase shifted from segment to segment so that the swimming wavelength is similar to the body length of the leech. (b) If the nerve cord is isolated in the dish, the absence of sensory feedback and mechanical coupling leads to a decrease in the segment-to-segment phase shift. The swim wavelength becomes larger than one body length. (c) Severing the nerve cord in an otherwise intact leech disrupts intersegmental coordination and leads to larger than normal phase shifts between segments on either side of the lesion. The swim wavelength becomes shorter than one body length. (Adapted with permission from Marder et al., 2005.)

recent model of respiratory pattern generation that includes sensory feedback to the respiratory CPG in the form of neurochemical signals such as blood carbon dioxide and oxygen content (Longobardo et al., 2005). The combined model reproduced normal breathing and a wide range of ventilation responses.

Systems in which the integration of neuronal, sensory, and motor components is well established include models of lamprey and leech swimming (Friesen and Cang, 2001). In both systems it has been shown that the interaction between central and peripheral mechanisms determines the oscillation period and oscillation phase shifts between different body parts and contributes to intersegmental coordination. Experimental findings regarding the effects of sensory feedback from the body wall and mechanical coupling between body segments on leech swimming are schematically illustrated in figure 12.6. A closed-loop model of the swim CPGs in each segment, their intersegmental connections, the mechanical interactions among muscles in neighboring segments, and the sensory feedback onto the CPGs replicates all the experimental results summarized in the figure (Cang and Friesen, 2002). Similarly, an integrated model of muscles, body mechanics, and water forces, and sensory inputs from stretch receptors onto the spinal circuits controlling locomotion in the lamprey was able to generate realistic and stable swimming, and pitch turns and rolls (Ekeberg and Grillner, 1999).

12.6 Choice of Modeling Software

The choice of a software package for developing a model is a difficult step. Often this choice is dictated by personal and local expertise with a particular package or in computer programming. Many researchers will simply write their own code or will use general programs for equation solving, like Matlab (<http://www.mathworks.com/>), or specifically for dynamic systems analysis, like XPPAUT (see the software appendix). The leech heartbeat CPG models have been implemented in different ways over the years, from self-generated code to different packages. Currently the CPG model is implemented in GENESIS (see the software appendix), an open-source general simulation package with near-infinite expandability owing to its object-oriented nature. Other open-source, widely used and supported packages are NEURON (see the software appendix) and SNNAP (see the software appendix). Each of these packages has its strengths and limitations. A tutorial based on the leech heart interneuron network described in this chapter is available in GENESIS.

Acknowledgments

We would like to thank Dirk Bucher for help with a figure. Support from the Burroughs Wellcome Fund to Astrid Prinz in the form of a Career Award at the Scientific Interface is gratefully acknowledged.

13

Large-Scale Network Simulations in Systems Neuroscience

Reinoud Maeijer, Michiel Berends, and Hugo Cornelis

13.1 Introduction

Modeling the function of large-scale neuronal systems often entails building and simulating a huge, heterogeneous network of neurons and analyzing its output. In such large-scale networks, each model neuron no longer corresponds to a uniquely identifiable nerve cell of the actual preparation, as it commonly does in small-network models (chapter 12; Getting, 1989). Instead, the model neurons are representative members of classes of cells sharing certain anatomical and physiological properties, and the imposed heterogeneity reflects the observed biological diversity in a statistical sense.

Many simulation packages provide powerful commands for multiplying template neurons or replicating small network modules along with their connections, each component being modeled as described in previous chapters. Hence except in those ultrarealistic networks where each modeled component is unique (chapter 9; Markram, 2006), implementing large networks does not pose, at first glance, many difficulties beyond those described for small networks (chapter 12).

A major methodological challenge then is to master the complexity of such networks by deciding which level of detail each component should be assigned. Too little as well as too much detail may be harmful. A network with little biological detail may have difficulty in proving its relevance for the experimental observation to be explained. Too much detail, on the other hand, may enhance the number of poorly constrained parameters so as to generate a level of freedom with which any desired phenomenon may be reproduced, obscuring again the critical biological processes and their underlying dynamics.

An indispensable, but often hidden, part of each network model therefore is the underlying scientific question: What was this model built for? The scientific question guides the construction of the model, the choice of solution method, and the strategy for analysis of the computed output. Starting from too vague a scientific question often only postpones problems to the stage of output analysis. Hence, network

models are not only implementations of available experimental data, it is safe to admit they are as much quantitative hypotheses about network functioning (Weizenbaum, 1976).

Instead of presenting a detailed recipe of how to build and implement a particular network (see, for instance, M. Wilson and Bower, 1989), we chose to address in this chapter generic methodological issues that arise during every modeling project. In this way we follow the natural sequence of the modeling process, from scientific question, over network design and implementation, to the analysis of datasets.

Section 13.2 provides examples of scientific questions that warranted the simulation of large-scale networks, along with a summary of alternative solution strategies that avoid large-scale simulations. These latter analytical methods are often invaluable intermediate steps in exploring the behavior of the full-blown network.

Section 13.3 deals with the most important and, often unknowingly, most time-consuming part of the modeling process: network design. The outcome is a complete mathematical description, including a list of model parameters and their allowable ranges. In particular, we try to evaluate the expected price and gain of omitting and adding certain biological detail. Implementation problems such as tailoring models to the available computing resources and numerically calculating their behavior are dealt with in section 13.4.

Section 13.5 briefly deals with the analysis of large-scale datasets and the epistemological question of how much physiological inference can be made from incompletely constrained models. Beyond this, large-scale simulation projects continually elicit new software development. Section 13.6 reviews current tools for the specification of multiparameter network models and for parallel processing. Finally, section 13.7 is a quick reminder of principles governing network dynamics. Throughout the chapter, reference is made to papers that demonstrate in an instructive manner the use of particular methods, preferably in an experimentalist's context.

13.2 Why Model Networks, and How Can One Avoid Large-Scale Network Simulations?

Generally speaking, two extreme types of questions can motivate researchers to turn to network modeling (figure 13.1a,b). In the first type, an observed input-output relationship is given, albeit often incomplete, in the form of a receptive field, an experimental recording, or a presumed computational function (such as stereo vision), and the question arises of what kind of network generates this stimulus-to-response transformation. Illustrative of this approach are numerous models built to solve the long-standing debate as to whether thalamocortical (forward) or intracortical (feedback) interactions are most constructive in shaping the receptive fields of simple and complex cells in the primary visual cortex (Wörgötter and Koch, 1991; Tao et al., 2006).

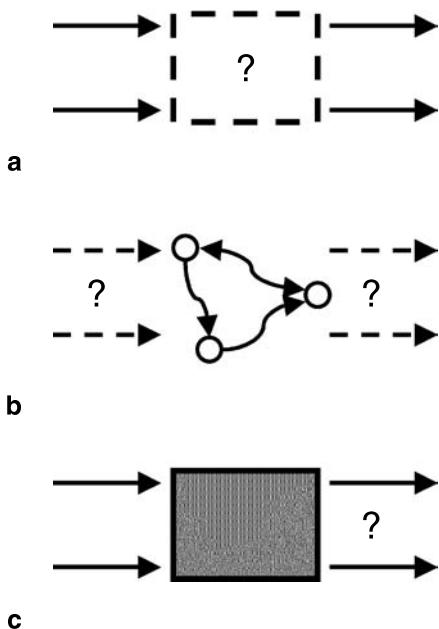


Figure 13.1

Three kinds of network modeling. The question marks refer to the unknown variables of interest. (a) The available data are a set of input-output transformations, but the circuit is unknown. (b) The network and its components are anatomically and physiologically well characterized, without exact knowledge of input and output. (c) Here the goal is to characterize the input-output transformation in pure mathematical form as a series of functionals without regard to its neuronal implementation (Poggio and Reichardt, 1976; Marmarelis, 2004).

At the other extreme (figure 13.1b) are networks in which the component neurons and synaptic connections are specified, but without presumptive evidence about their input-output relationship, raising the question of what task the network is actually performing. Illustrative here is a model of neocortex that attributed improved signal processing to the incorporation of lamina-specific connectivity and short-term plasticity (Häusler and Maass, 2007).

Obviously, modeling studies can be more successful when details about both the circuit and the input-output relationship are known, as for instance is the case for spinal cord (chapter 12). For other systems, located many synapses away from the sensory surfaces and the effectors, a functional characterization in terms of spike train computations may be more feasible. It is worth mentioning in this respect that a separate modeling approach specifically aims at a complete systems characterization in terms of mathematical input and output relations, often obtained during white-noise stimulation (Marmarelis, 2004; figure 13.1c).

Why Large-Scale Simulations of Biologically Realistic Networks?

Before embarking on a long-term modeling project, two questions must be addressed. First, why should one simulate huge numbers of neurons and synapses? Second, should each neuron (and synapse) be represented with all the available detail? (and not just as, for instance, leaky integrate-and-fire devices; chapter 7). The straightforward answer, which is to say that this is the way the brain is built, is too naive, for more efficient strategies can exist to solve well-defined scientific questions, as will be illustrated shortly.

As noted earlier, scientific questions determine from the outset the network parameters and outcome variables of interest. Large and more detailed networks, by having more free parameters, can obscure the results instead of clarifying them. Sometimes a new approach, such as taking into account the intrinsic stochastic nature of bioelectrical phenomena, can be more important than the level of detail (for a similar view from molecular biology, see Vilar et al., 2003). Reasons then for building networks with large numbers of neurons and synapses often fall into one or more of the following categories.

Dynamics

Feedback or lateral connections (for an explanation of terms, see section 13.7) abound at almost every level of the nervous system. So even if the sole outcome of interest is the response of a single neuron, say, a simple cell in the primary visual cortex, the resulting intrinsic cortical dynamics cannot be disregarded. Given that every cortical neuron receives thousands of synapses, mostly of cortical origin, a network of such large scale arises that it will even be necessary to consider strategies to dilute the neuron population, to reduce network size, and to minimize boundary effects (see sections 13.3–13.4).

Representation

Many models acquire their size by the desire to explain macrophenomena or to represent high-dimensional input spaces. Examples are anatomically refined models of wave propagation in motor or epileptic activity, or studies of topographic map formation and self-organization (Kempter et al., 2001). As a more explicit example, the representation of the time elapsed after stimulus onset was found to improve with increasing numbers of granule cells in a network model of the cerebellar cortex (Yamazaki and Tanaka, 2007). Models of memory capacity evidently require sufficient numbers of synapses for storing stimulus representations.

Statistics

Networks containing large numbers of neurons and synapses reduce sampling artifacts that could arise, for instance, when the connectivity between neuron classes is known only as a probability distribution, or when the output variable is a spatially

averaged quantity like the electroencephalogram (EEG) signal (Esser et al., 2007). It will further be shown that the very number of synapses each neuron receives can affect network dynamics (section 13.3). Finally, an intriguing question is whether there are phenomena that occur only in networks that exceed a critical number of neurons or synapses.

Analytical and Alternative Approaches

Many scientific questions concern the potential effect of a few biological parameters on a single output variable. According to the desired scale of resolution in space and time, analytical methods can be more suitable for solving such questions. They can yield closed-form expressions for parameter dependencies that large-scale network simulations would be able to provide only after painstaking data analysis. Moreover, many insightful relationships were established analytically in often-forgotten literature from the precomputer era. For instance, the dominant impact that connection delays have on the oscillation frequency of inhibitory networks, recently observed through large-scale network simulations (Brunel and Wang, 2003; Maex and De Schutter, 2003), had in retrospect been derived analytically by the group of Hartline (Ratliff et al., 1969).

Common strategies for casting biological problems into analytically treatable expressions are to reduce the number of equations and variables, to limit the associated range of time scales, to convert the differential equations into algebraic equations, and to linearize the neurons' input-output functions (chapter 1, section 3). For instance, if slow network oscillations are the phenomenon of interest, faster processes like spike generation may be assumed to average out among the individual neurons, justifying the use of the mean network firing rate as the single, lumped output variable (Marchetti et al., 2005).

Mathematical Description

We illustrate some of these principles starting from the formal representation of a neuron, in which each compartment is modeled as a resistor-capacitor (RC) circuit having as parameters the membrane capacitance c , the leak conductance g_{leak} and leak potential E_{leak} , and as output variable the membrane potential V_i . Then,

$$c \frac{dV_i(t)}{dt} = g_{\text{leak}}[E_{\text{leak}} - V_i(t)] + I_i^{\text{intrinsic}}(t) + I_i^{\text{dend}}(t) + I_i^{\text{applied}}(t) + I_i^{\text{network}}(t) \quad (13.1)$$

$$I_i^{\text{network}} = f(V_1, \dots, V_i, \dots, V_n). \quad (13.2)$$

The currents in the current-balance equation (13.1) pass through (from left to right) the capacitive branch of the membrane, the (passive) membrane resistance, the (active) membrane channels, the axial dendritic or axonal cytoplasm, an experimen-

tally applied electrode if present, and the synapses made by other neurons in the network. Only this last term makes a network model different from single-neuron models (e.g., equation 11.1).

Equation 13.2 clarifies that the synaptic input ultimately may depend on the membrane voltage V_i of any other compartment in the network (for instance, through gap junctions). Other equations, such as those describing the kinetics of the gating variables of ionic channels and their associated parameters, have been omitted. Of course, there should be as many equations as variables. In the remainder of this section we consider only single-compartment neurons, which are isopotential and lack the axial current component.

Firing-Rate Models

A first simplification often applied in analytical network models is to regard each output variable V as denoting the average firing rate of a population of similar neurons, instead of representing the membrane potential of a single neuron or compartment. Obvious gains are a reduction in the number of output variables, the dismissal of fast processes underlying action potentials, and (partly) the loss of nonlinearity in the spike threshold. Such rate models can be used as components in a coarse-grained synaptic circuit. For instance, H. R. Wilson et al. (2001) investigated whether waves traveling across the visual cortex could account for the perceptual propagation of ocular dominance along either of two stationary stimuli in a binocular-rivalry task. To explain this phenomenon, which operates on a time scale of seconds and involves large parts of the visual field, they represented the visual cortex as two chains of (monocular) model neurons, competing through inhibitory interneurons.

Field Representation

Alternatively, neuron populations can be represented as continuous fields (Cowan, 1974), raising neuron numbers to infinity but reducing the mathematical description to a system of partial differential equations:

$$c \frac{\partial V(x, t)}{\partial t} = g_{\text{leak}}[E_{\text{leak}} - V(x, t)] + I^{\text{intrinsic}}(x, t) + I^{\text{applied}}(x, t) + I^{\text{network}}(x, t) \quad (13.3)$$

$$I^{\text{network}}(x, t) = \int_{-\infty}^{\infty} dx' \int_{-\infty}^t dt' k(x - x', t - t') V(x', t'). \quad (13.4)$$

The new independent variable x commonly represents neuronal or receptive-field position, but can denote other features as well, such as orientation preference (B. Blumenfeld et al., 2006) or temporal phase with respect to an oscillatory cycle. The uniform representation allows spatial and temporal interactions (for instance, the spatial extension of lateral inhibition, or the temporal waveform of postsynaptic currents) to be expressed as convolutions (see equation 13.4).

For linear networks, convolutions can be transformed into simpler algebraic operations (Ratliff et al., 1969; Shamma, 1998). For nonlinear models, the reduced set of partial differential equations can be solved qualitatively or quantitatively by means of numerical simulation packages like XPPAUT (see the software appendix).

Mean-Field Approximation

In a further simplification, the synaptic inputs each neuron receives from other neurons in the network can be replaced by a term representing the average activity of the entire network over a small time interval. This amounts to reducing the network to the activity of a single, representative neuron:

$$c \frac{dV(t)}{dt} = g_{\text{leak}}[E_{\text{leak}} - V(t)] + I^{\text{intrinsic}}(t) + I^{\text{applied}}(t) + I^{\text{network}}(t) \quad (13.5)$$

$$I^{\text{network}} = f(V). \quad (13.6)$$

In studying fast processes, slow variables can be taken as constant parameters, whereas the study of slow processes can take advantage of averaging out fast processes. For instance, Marchetti et al. (2005) used a mean-field approximation to identify the nonlinear biophysical mechanisms causing periodic transitions between states of distinct spike rates in the developing spinal cord. As such, temporal resolution and spatial anisotropies were sacrificed to focus on the mean firing rate of a typical neuron during slow variations in synaptic efficacy and ionic concentration.

To close this section, analytical models turn out to be invaluable in deriving parameter dependencies and explaining network dynamics. More detailed models can be indispensable, however, in identifying the biological components involved, so as to be able to conclude, for instance, whether the source of a delay underlying network oscillations is to be found in axonal spike propagation, activation of slow metabotropic receptors in synapses, or feedback excitation; or whether a nonlinearity involved in stimulus selectivity is caused by firing-rate adaptation, short-term synaptic depression, or shunting inhibition.

13.3 Designing a Large-Scale Network Model

When a scientific question requires that a network model be built from detailed biological data, so that each of its parameters roughly corresponds to an actual biological component of the circuit and expresses quantities in similar units, the next decision to be made is what available data should be included and what may be neglected. As noted earlier, no fixed heuristics are available to guide us. This central section therefore aims to give a flavor of which details may matter for particular scientific questions.

Defining Neuron Classes

In most large-scale network models, neurons are representative units that acquire their characteristics from the class to which they belong. Neurons of the same class have the same dynamics, release the same transmitter, and follow the same connectivity patterns. Biological diversity among members (intraclass variability) is introduced afterward through randomization of the parameters of a template neuron (see the subsection *Heterogeneity and Noise*).

Delineating neuron populations already implies a simplification because classical dichotomies between cell types (for instance, between stellate and basket cells in the cerebellar cortex; Sultan and Bower, 1999) often are mere distinctions between extreme examples of what is actually a continuum (for a methodological discussion, see Sherman and Guillery, 2006). Moreover, in experimental studies, the passive and active intrinsic properties of neurons appeared *not* to be invariant, but rather flexible (Steriade, 2003). The distribution of neurons across different classes depended on the experimental conditions, and even classification of the same neuron could change according to the strength of current injection or the state of vigilance. Hence a model aspiring to account for network behavior under various conditions should also incorporate these modulators.

Some earlier network models oversimplified the distinctions between neurons based on presumed symmetries, for instance, by assuming that inhibitory neurons have responses identical to those of companion excitatory neurons (which allows discarding the inhibitory population altogether; Xing and Gerstein, 1996) or that inhibitory and excitatory neurons receive identical inputs (Maex and Orban, 1996). However, if Dale's principle of unipolar connections is violated, correlations and fluctuations in the time structure of activity may be lost (Kriener et al., 2008).

Connecting the Circuit: Gross Connectivity between Classes

At this early stage, a simple wiring diagram in which all neurons of the same population are lumped into a single unit and the connections between units indicate the gross influences that populations exert on one another, can give indications of the expected dynamics and warn of potential pathological deviations.

It should always be kept in mind that networks acquire intrinsic dynamics from their architecture and connectivity, which are largely independent of the details of their component neurons and synapses (I. Stewart, 2004), although connectivity itself may also be subject to modulators. Actually, some of the earliest successes in computational neuroscience were elaborations of simple wiring diagrams implementing an engineering concept: lateral inhibition for contrast enhancement in the retina (Hartline and Ratliff, 1972), delayed correlation for detection of visual motion (Poggio and Reichardt, 1976), lateral inhibition between units receiving complementary (push-

pull) inputs for temporal integration by the oculomotor system (S. C. Cannon and Robinson, 1985), and Jeffress' countercurrent circuit for detection of interaural coincidence (for recent implementations, see Kempfer et al., 2001 and Y. Zhou et al., 2005).

Since the connection patterns of most actual networks are described incompletely, at best in a statistical sense, it is important to keep the connectivity parameters under constant control during all stages of the modeling process. Section 13.7 gives practical rules that can help predict the effects of parameter choices, such as connection strength and delay, on network behavior.

Here we briefly treat the problem of stability: how to prevent neurons that excite each other from becoming all recruited and driven to their maximum firing rate (Griffith, 1963). It is noteworthy that most biological networks possess stringent controls because they operate over wide dynamic ranges with invariant response properties, indicating that their gain is under the constant influence of (neuronal and synaptic) homeostatic mechanisms. However, so long as the exact nature of these mechanisms remains uncertain, model networks need careful design and parameter tuning.

A straightforward solution to the stability problem in network models is to balance feedback excitation with feedback inhibition. It was found that lateral inhibition extending over longer distances than the range of local excitatory connections stably limited the spread of self-excitation (Xing and Gerstein, 1996). There is, however, a limit on the strength that feedback inhibition can take. Lateral inhibition amplifies heterogeneity in the input, and when it is too strong it results in a winner-takes-all behavior (section 13.7). This nonlinearity can lead to hysteresis if after cessation of the stimulus the winner has to passively leak to its subthreshold potential, preventing competing neurons from becoming responsive (a kind of dynamics that may nevertheless be useful in slower processes like developmental self-organization).

A variety of strategies have been proposed to keep the network optimally responsive to external inputs in the presence of feedback excitation and inhibition. Push-pull amplification, in which a neuron responding to the onset of a stimulus receives inhibition from a complementary neuron excited by stimulus offset, speeds extinction of the winner's activity. Similar effects can be achieved by adding purely feedforward inhibition or by using neuron models with firing-rate adaptation (Holt, 1998). Divisive inhibition through high-conductance shunting synapses (see section 13.7) is less competitive than subtractive inhibition and prevents undue amplification of network heterogeneities (Vida et al., 2006). Incorporating short-term synaptic plasticity (Rothman et al., 2009), or synaptic failure (Tsodyks and Sejnowski, 1995), can self-regulate the level of excitation. Several studies demonstrated that balanced but sparse connectivity limits the inherent instability of feedback excitation while preserving optimal responsiveness of the neurons (Tao et al., 2006; Vogels et al., 2005).

Finally, it is worth mentioning that layered feedforward networks (such as synfire chains; Abeles, 1991) may suffer related problems of reliable and stable transmission of spiking activity. With inappropriately tuned parameters, the signal may die out (Diesmann et al., 1999) or ignite the surrounding network (Vogels et al., 2005).

Modeling Component Neurons

Determining which features of network dynamics originate from connectivity parameters and which from neuronal or synaptic parameters can be difficult in experimental and modeling settings alike. From a computational point of view, most neurons are threshold devices that integrate myriad inputs in their dendrites in a nonlinear fashion, that are capable of generating spikes with a submillisecond precision, and that display intrinsic resonance. Which of these properties should dominate the neuron model depends on the scientific question. If the propagation or synchronization of activity across a huge network of identical units is studied, simpler components like flip-flops (Lewis and Rinzel, 2000) or integrate-and-fire devices can yield instructive results (Brunel and Wang, 2003; Diesmann et al., 1999; van Rossum et al., 2002).

At the other extreme of complexity are neuron models having intrinsic dynamics capable of determining the frequency of local field potential oscillations (Destexhe and Sejnowski, 2003; Traub et al., 2005; Llinás et al., 1991) or even the time course of behavioral phenomena like motor kinetics (Karameh and Massaquoi, 2005). In this and the following subsection, we briefly discuss the virtues and potential side effects of particular features of model neurons and synapses.

Intrinsic Membrane Excitability

Leaky-integrate-and-fire (LIF) point neurons facilitate analytical treatment (Koch, 1999; see chapter 7, section 3). Recent modifications, such as the quadratic LIF model and the incorporation of firing-rate adaptation, have improved the accuracy of spike timing (Jolivet et al., 2006; chapter 7, section 4). The spike response model generalizes the LIF model by allowing for other synaptic impulse response functions than the exponential decay kinetics of the membrane's RC circuit. Much more refined membrane models are needed, for instance, to interpret and predict the effects on network dynamics produced by pharmacological or genetic manipulation of individual ionic currents (Destexhe and Sejnowski, 2003; Traub et al., 2005; chapters 11 and 12).

For a proper interpretation of network output, it is highly recommended that for each model neuron (whether it be custom-made or downloaded from a database) a frequency-intensity (f - I) and a phase-response curve be constructed and the model tested for intrinsic resonances. The f - I curve plots the mean firing rate f during injection of a constant current I of varying magnitude. The neuron's input resistance and

total membrane capacitance will greatly determine the threshold current and the slope of the f - I curve, respectively (Koch, 1999). (Note that most simulators, using the specific membrane capacitance and resistance as parameters, automatically scale intrinsic and synaptic currents to membrane area, allowing neuron size to change without further measures.) Characteristics of interest in the responses to current injection are (1) the presence of distinct bursting and regular-firing modes (although model neurons that “burst” during somatic current injection may fire regularly when they are excited synaptically by a network); (2) rebound depolarization, which can stabilize rhythmic network behavior by discharging the neurons after each phase of inhibition; and (3) adaptation and saturation of the firing rate, which introduce additional time constants and nonlinearities, with secondary effects on synchronization (Crook et al., 1998; van Vreeswijk and Hansel, 2001; Borgers et al., 2005).

Of particular importance also is the neuron’s phase-response curve (Rinzel and Ermentrout, 1998) which plots, for a test pulse applied at varying phases in the firing cycle, the advancement or delay of the next spike. When an excitatory pulse is applied shortly after a regular spike, some neurons (called type II; chapter 7, section 4) paradoxically delay their next spike. This phenomenon can stabilize synchronous oscillations because a neuron firing too fast will receive, early in its firing cycle, pulses from properly timed neurons and correctively delay the timing of its next spike.

The nonlinear voltage-dependent gating of ionic channels and the delayed interplay between opposing currents endow model neurons with an intrinsic resonance (Buzsáki, 2006). This resonance can manifest itself by subthreshold oscillations during constant-current injection (chapter 7), or as a tuning of response amplitude to frequency-modulated sine wave currents (Hutcheon and Yarom, 2000), or as an increased regularity of firing at particular levels of noisy excitation (coherence resonance).

Morphology

Apart from the modeler’s desire to accurately reproduce the layered architecture of a circuit (chapter 9), local nonlinear interactions constitute the most important reason for building multicompartmental dendrites. These nonlinearities can take several forms, such as shunting inhibition (see later discussion), saturation of the postsynaptic potential (Solinas et al., 2006), local interactions between ligand- and voltage-gated currents, clustering of ionic channels and buffers, or electrical coupling through gap junctions (Migliore et al., 2005b). Other position effects that may demand a spatial representation of the dendrite are gradients in the density of voltage-gated channels (chapter 11) or gradients in the reversal potential or synaptic kinetics of ligand-gated channels.

Although axons also play a major role in network dynamics (through spike initiation, propagation delay and failure, routing, electrical coupling, etc.), most neuron

models used in large-scale networks contain at most a few axon compartments (Traub et al., 1999b). Unmyelinated axons were modeled recently based on patch-clamp data (Alle and Geiger, 2006; Shu et al., 2006), but their profuse branching requires a multicompartment representation that at present impedes standard use in network simulations. McIntyre et al. (2004) provide a detailed model of an unbranched myelinated axon of a central neuron. Complete morphological reconstructions are rare, but Binzegger et al. (2005) developed a random-branching algorithm that statistically reproduced actual axonal arbors.

Obviously, there is a tradeoff between the complexity of neuron models and the affordable network size. However, recent methods for parameter estimation, designed to reproduce a desired electrophysiological profile (chapter 2), will facilitate the conversion of detailed single-neuron models to reduced models more suitable for large-scale networks.

Modeling Component Synapses

Most neurons communicate through action potentials and thus are pulse-coupled devices. Conceptually, synapses are part of the recipient neuron. The classic manner of implementing synaptic communication in large-scale networks is by representing the afferent spike as an impulse that evokes in the target compartments a response (of conductance, current, or voltage) with predefined delay, amplitude, and waveform (for instance, a sum of exponentials, equation 6.4). Intermediate steps are typically neglected, including axonal spike propagation (and axons altogether, see earlier discussion), transmitter release and diffusion, and overspill of transmitter from neighboring synapses, all of which may contribute to the variability of the unitary postsynaptic response.

Synaptic Strength

Evidently, synaptic strengths are among the most important parameters in network models. Even in cases where the unitary postsynaptic currents or potentials have been fully described, complete information about the total number of afferents and their firing rate is mostly lacking. Given that in addition most network models necessarily are rarefied (section 13.4), proper parameter values often must be assessed from surrogate measures like the resultant level of activity in the target population. In a later subsection (Synaptic Density) we illustrate the additional effect that varying the number and strength of synapses can have on spiking statistics. Variability of the postsynaptic response at individual synapses is usually not modeled, although adapting the probability of synaptic failure was found to be an efficient way to compensate for changes in synaptic number after rescaling of the network (Tsodyks and Sejnowski, 1995).

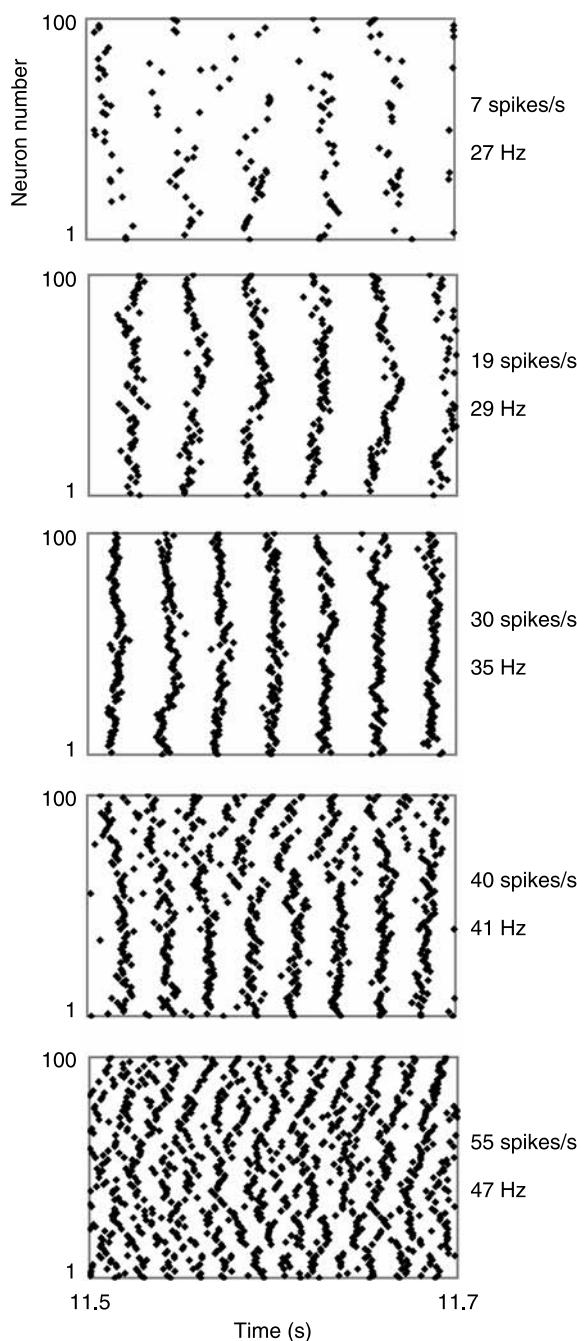
Synaptic Latency

Axons act as delay lines and are frequently modeled as such. In small networks (chapter 12), delays can be insignificant. Indeed, dual recordings of nearby neuron pairs typically report latencies not greater than a millisecond. When the spatial scale of the network increases, delays are expected to measure several milliseconds, although exact values are scarce. No matter how small, delays cause resonance effects by restricting interneuronal spike synchronization to well-determined frequency bands. Indeed, under certain conditions, delays paradoxically stabilize synchronization (van Vreeswijk et al., 1994). In networks with reciprocal inhibitory coupling, resonance emerges at a frequency of about $1/(4d)$ (with d denoting the connection delay; Brunel and Wang, 2003; Maex and De Schutter, 2003), whereas the optimal firing rate for synchronization of a network with reciprocal excitation equals approximately $1/d$. Often it is the *average* connection delay between neuron populations that matters, but this parameter is even more difficult to derive experimentally. Note that most analytical models use smooth, distributed delays that can be cast into a continuous kernel (Marchetti et al., 2005).

Figure 13.2 illustrates frequency-tuned synchronous oscillations obtained in an inhibitory network model with connection delays that was excited to a range of levels of activity by varying the firing rate of its afferent fibers. The complementary effect—the change of both oscillation frequency and average firing rate when the delay is varied—is shown for another network in figure 13.3. This latter network, which possessed stronger inhibition and greater heterogeneity, also makes it clear that the individual neurons can spike at rates much lower than the network oscillation frequency (Brunel and Wang, 2003).

Subtractive versus Divisive Inhibition

GABA_A receptor-activated currents can be hyperpolarizing, silent, or depolarizing, depending on whether the Cl⁻ reversal potential is more negative than, equal to, or more positive than the membrane potential. The silent or shunting case in particular has evoked computational interest as a biological implementation of a nonlinear operator (Poggio and Reichardt, 1976). “Shunting” synapses may selectively veto certain sequences of inputs by neutralizing the current through nearby excitatory synapses activated shortly before (Koch, 1999). Likewise, tonic levels of shunting inhibition can modulate in a nonlinear manner the gain of a neuron’s *f-I* curve (Holt, 1998; Berends et al., 2005 and references therein). Depolarizing GABA_A receptor currents have been reported to prevail not only in the developing but also in the mature brain. They tend to clamp the membrane potential at a level between the resting potential and the firing threshold, and were found more effective than hyperpolarizing currents in stabilizing the frequency of gamma oscillations in interneuron network models (Vida et al., 2006).



Synaptic Plasticity

There is an increasing appreciation of the impact that dynamic synapses (chapter 6, section 4) have in shaping a neuron's spiking pattern (Gerstner et al., 1993) and its receptive-field properties (Hosoya et al., 2005). Indeed, dynamic synapses can be thought of as rewiring the circuit temporarily (short term) or permanently (long term).

Short-term facilitation and depression have effects running into several hundreds of milliseconds, a time range beyond the realm of basic components like the neuronal cell membrane and most unitary synaptic currents. They greatly extend the "memory window" of a network (Abbott and Regehr, 2004; Häusler and Maass, 2007), and their capacity to generate apparent delays has been applied successfully in models of directional selectivity (Chance et al., 1998). Short-term depression of excitatory synapses may also contribute to gain control and contrast invariance because each rise of the afferent spike rate enhances the proportion of depressed, less-effective inputs (Rothman et al., 2009).

Long-term plasticity (chapter 6, section 6) causes the network to vary on longer time scales, driven by correlations in the input. Feedback circuits can have a stabilizing effect on long-term plasticity rules by self-regulating the afferent firing rate (Kenyon et al., 1998). However, plasticity acting simultaneously at different levels of the circuit may be unstable and erase learned patterns (Medina and Mauk, 1999; P. D. Roberts, 2007).

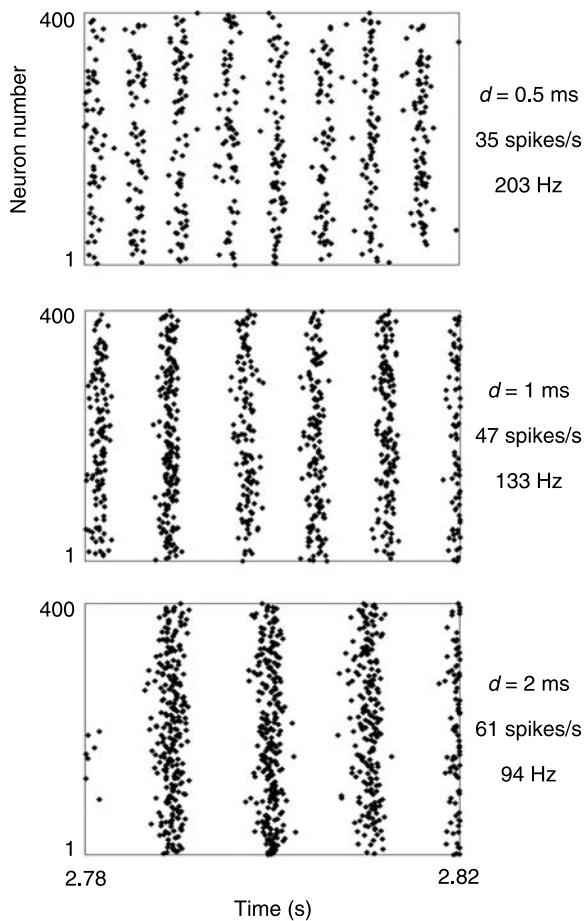
Besides synaptic strength, latency can also be learned in models of self-organization and map formation. Spike-timing-dependent plasticity has been successfully implemented as the selection of discrete axonal conduction delays from an available distribution (Izhikevich et al., 2004; Kempter et al., 2001).

Electrical Synapses

Gap junctions are usually modeled as pure conductors between dendritic compartments (equation 12.2), ignoring their potential gating by voltage or chemical modulators (for detailed models, see Vogel and Weingart, 1998). Their incorporation into

Figure 13.2

Frequency tuning of synchronous oscillations in a network of inhibitory neurons with delay connections. The one-dimensional network had a mean connection delay of 8.5 ms, each neuron making a synapse on all of its r th neighbors on either side ($1 \leq r \leq 16$) with a delay set to r milliseconds. The predicted optimal oscillation frequency was $1/(4 \times 8.5)$ or 29.4 spikes/second (Brunel and Wang, 2003; Maex and De Schutter, 2003). Individual IPSCs had a decay time constant $\tau = 12$ ms. The spike-time raster plots show snapshots of the network's firing pattern from five simulations using varying degrees of excitation; to the right are shown the mean neuronal spike rate (spikes/second) and the network oscillation frequency (hertz). Synchrony is poor when the network fires at slow rates (top); it gradually builds up to reach a maximum precision at the mean neuronal firing rate of 30 spikes/second, and again deteriorates when the level of excitation is further increased (bottom). Frequency tuning is more prominent in larger, two-dimensional networks (as shown in figure 13.3), but never occurs in inhibitory networks with instantaneous, nondelayed connections (van Vreeswijk et al., 1994).

**Figure 13.3**

Connection delays can strongly influence network dynamics. The three panels are 40-ms raster plots of spike timing of a lateral inhibitory network (400 inhibitory neurons randomly coupled via somatic GABA_A receptor synapses, with a probability $p = 0.2$; see Maex and De Schutter, 2003, for further details). To the right are given the mean neuronal spike rates (spikes/second) and the network oscillation frequencies (hertz). The three networks were identical except for the delay over the connections: the latency d of the IPSCs measured 0.5 ms (top), 1 ms (middle), and 2 ms (bottom). These small differences in delay affected not only the network oscillation frequency but also the average neuronal spike rate. Note that the emerging oscillation frequency was attracted toward the delay-induced resonance frequency of about $1/(4d)$.

network models has a phase-locking effect upon output spike trains, and traveling waves may self-organize in spontaneously active networks (Lewis and Rinzel, 2000). As such, electrical coupling enables the component neurons to recruit and synchronize their intrinsic oscillators, and this presumably underlies network oscillations in nonsynaptic circuits (Llinás et al., 1991; Steriade et al., 1996; Roopun et al., 2006). Adding gap junctions also tends to homogenize a network, which is illustrated in figure 13.4, where synchronous oscillations lost after enhanced parameter randomization were restored with weak coupling. Recent models also postulate a role for axo-axonal gap junctions in the induction of very-high-frequency (200 Hz) ripple oscillations (Traub et al., 1999b; Maex and De Schutter, 2007 and references therein).

Connecting the Circuit: Fine Connectivity between Neurons

Once the network has been populated with neurons, each at a unique position, the individual connections are to be laid. Connectivity is usually described in a statistical sense as a distribution or kernel, the shape and extent of which contribute to network dynamics (Shamma, 1998). The connection probability, along with the density of neurons, also determines the synaptic density. Contrary to intuition, the absolute number of synapses each neuron receives will often be more influential for network behavior than the connection probability.

Connection Topology

The actual local and long-range connectivity can be derived from dual recordings and tracer studies, and databases provide metainformation and advanced tools for visualizing and extracting data (Bjaalie et al., 2005; Dyhrfjeld-Johnsen et al., 2005; Kötter, 2004). For most networks, however, the exact topology of the connections remains unknown, though recent experiments strongly suggest that actual connectivity is more specific and less random than that commonly applied in large-scale network models (Song et al., 2005; Y. Wang et al., 2006).

At a descriptive level, and building on graph theory, topology can be characterized by metrics like clustering (the degree of local coupling) and path length (the average length, in number of synapses, of the shortest path between any pair of neurons). For instance, it has been argued that the neocortex has a “small-world” connectivity (Sporns, 2006 and references therein) in which clustering (or map formation) minimizes wiring length, whereas the sparse long-range connections diminish the average path length. Functionally, such small-world topology is thought to enhance information processing and to increase the speed of synchronization (Häusler and Maass, 2007; Percha et al., 2005; Calcraft et al., 2008).

In addition, slight anisotropies in connectivity can give rise to traveling waves (M. Wilson and Bower, 1992; Ermentrout and Kleinfeld, 2001; Shamma, 1998), or

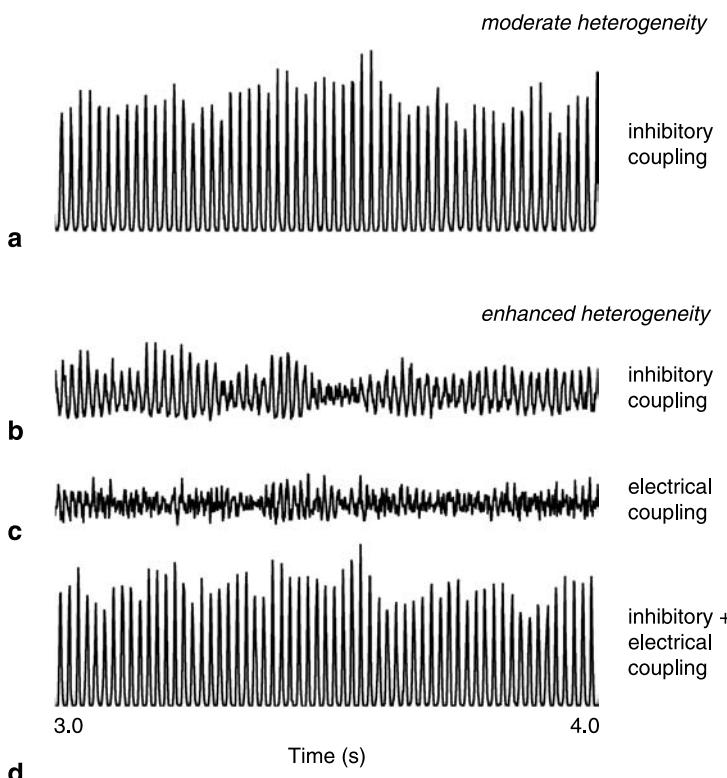


Figure 13.4

Electrical coupling enhances network homogeneity. Illustrated are four simulations of a two-dimensional network of 625 six-compartmental inhibitory neurons. The traces are population spike-time histograms (bin width 1 ms; total trace duration 1 s) as surrogate measures of the local field potentials. In a the network had lateral inhibitory connections of moderate heterogeneity (default parameters as in Maex and De Schutter, 2003). The strong oscillations of network a were abolished, as shown in b, after the network's heterogeneity had been enhanced by extending the randomization interval of the synaptic weights. Adding, in this heterogeneous network, six 1-nS gap junctions to the proximal dendrite of each neuron (so as to connect each neuron to its six neighbors) sufficed to restore the oscillations, as shown in d. The control trace in c demonstrates that the weak gap junctions failed to synchronize the network in the absence of lateral inhibitory coupling. From an experimentalist's point of view, b and c would represent the abolishment of the oscillations present in d by blocking gap junctions (b) or GABA_A receptors (c).

receptive-field asymmetries like directional selectivity (Poggio and Reichardt, 1976). The connection pattern further influences the speed of wave propagation (Traub et al., 1999b; Idiart and Abbott, 1993; Lewis and Rinzel, 2000), an important parameter in studies of epilepsy.

From a practical point of view, given that an explicit axon representation is mostly absent in network models, it is common use to lay connections directly from the spike-initiating site (often the somatic compartment) to potential targets within a cubic or ellipsoid volume, and to repeat this procedure for each neuron in the network. It is worth noting here that many simulation packages, such as GENESIS, provide powerful connection commands that use templates for the spatial distribution of the efferent connections as well as specifiers for the distribution of synaptic strengths and delays (see the software appendix).

Connection Density

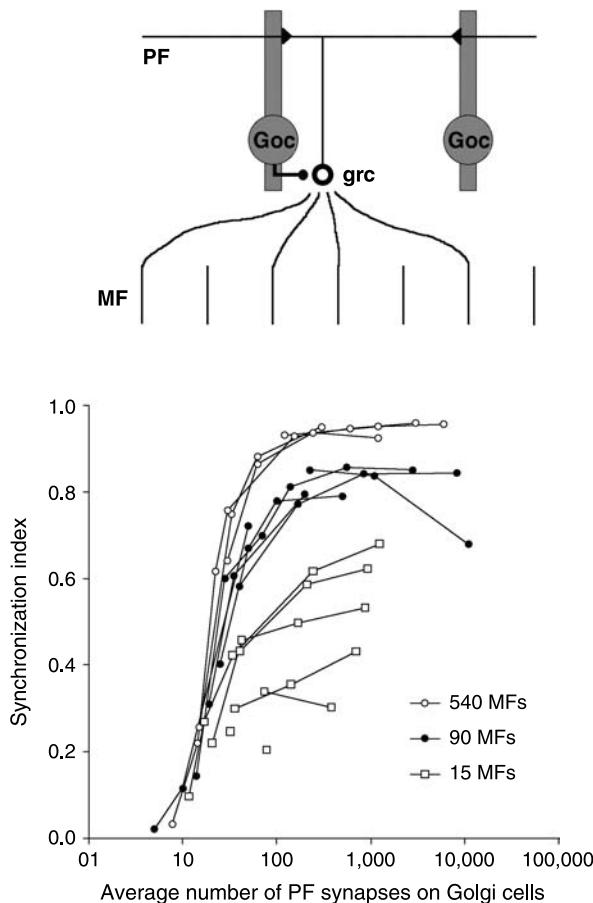
As noted earlier, most biological networks appear to be sparsely connected (Laughlin and Sejnowski, 2003), and the behavior of many network models is also remarkably robust to rarefaction of the synapses (Börgers and Kopell, 2003). Any observed effects of connection density are often due to changes in the absolute number of synapses each neuron receives. Decreasing the number of synapses (and enhancing their strength in reverse proportion to keep the mean activity levels approximately constant) increases the amplitude of the excursions the membrane potential makes in the postsynaptic neuron. Because neurons are threshold devices, the probability of spike generation rises as well. As a result, network synchronization may decline, even though this property is almost completely invariant to changes in connection probability (the relative synaptic numbers), as illustrated in figure 13.5 for a model of the cerebellar granular layer (Maex and De Schutter, 1998).

Heterogeneity and Noise

Network Heterogeneity

Because networks are assembled starting from template neurons, synapses, and connection patterns, it is important to introduce heterogeneity as a reflection of biological diversity, although the degree of heterogeneity is poorly constrained by the literature. See chapter 12, section 3 for a discussion of natural variability in intrinsic neuronal properties and the consequential nonuniqueness of parameters.

Heterogeneity is indispensable for breaking network symmetry (and hence preventing deadlocks), and it extends a network's processing capacity or the number of different states it can take (Eliasmith and Anderson, 2003). Among the many forms of embedding heterogeneity are variations in neuronal morphology and stochastic connectivity, and randomization of synaptic strengths and neuronal resting potentials. Alternatively, it can be applied as gradients to externally applied currents or as external noise.

**Figure 13.5**

The effect of network rarefaction on the power of synchronous oscillations in a model of the granular layer of the cerebellar cortex. Upper schematic: network layout. A fixed population of 30 Golgi cells (Goc) exerted feedback inhibition on a granule cell population (grc) of variable size. Also, the size of the pool of mossy fibers (MF, the external afferents to granule cells) was varied, as was the connection probability of granules to Golgi cells via parallel-fiber (PF) synapses. The lower plot shows the relation between the absolute number of PF synapses on each Golgi cell and network synchronization. The curves were constructed by varying the connection probability from PFs to Gocs in networks with 15 MFs and 45, 105, 462, 810, 990, 1,155, or 1,365 grcs; 90 MFs and 87, 345, 855, 4,662, 13,365, or 17,655 grcs, or 540 MFs and 537, 2,145, 5,355, or 10,695 grcs. Each mossy fiber generated spikes from a stationary Poisson interval distribution. The network oscillations were quantified by a synchronization index (a value of one corresponding to perfect synchronization and periodicity and zero indicating an asynchronous state; see Maex and De Schutter, 1998). Note that although synchronization improved with increasing numbers of mossy fibers and granule cells, and with increasing connection probability, the critical parameter was the absolute number of PF synapses on individual Golgi cells.

External Noise

Noise here denotes temporal randomness added to neurons, synapses, or afferent fibers. By continually “shaking” the neurons, noise prevents the network from getting locked in an exceptional state following a particular choice of initial conditions. Noise is often part of the input (see next subsection), or can be implemented in neurons by using stochastic rather than fixed spike thresholds (e.g., Jolivet et al., 2006), by applying ectopic axonal spikes (Traub et al., 1999b), or by using stochastic differential equations for the kinetics of the gating variables of ionic channels (Saarinen et al., 2008).

An equally important function of noise application is to compensate for the restricted size each network model inevitably has, and hence for the input originating from neurons omitted from the model. These omitted inputs affect neuronal dynamics in several ways in the integral network (Fellous et al., 2003). By enhancing the membrane conductance, they accelerate the neuron’s dynamics (Bernander et al., 1991), and by bringing the average membrane potential closer to threshold, they allow small fluctuations, potentially brought about by a single afferent spike, to fire the neuron. This latter stochastic resonance effect enables, for instance, feedforward networks to transmit asynchronous firing-rate information, whereas in the absence of noise, only packages of synchronous spikes would cross the firing threshold and propagate to the next layer (van Rossum et al., 2002).

The effects that noise exerts on the membrane time constant and the average voltage can be manipulated independently in models if the ratio of excitatory to inhibitory synaptic currents is kept approximately constant. Such so-called balanced networks have more realistic response properties, such as fast reaction time, optimal gain control, and increased signal-to-noise ratio (van Vreeswijk and Sompolinsky, 1996; Vogels et al., 2005; Tsodyks and Sejnowski, 1995; Koch, 1999).

Noise is often modeled as an Ornstein-Uhlenbeck process (introduced in physics to model the velocity of a Brownian motion particle; see Tuckwell, 1995; Fellous et al., 2003), although the high intertrial variability of stimulus-induced responses in several cortical areas would argue for stochastic inputs with even higher-order correlations (Jolivet et al., 2006 and references therein).

Modeling the Input

Usually the input is a set of afferent fibers, although sometimes currents (Vida et al., 2006) or conductances (Harsch and Robinson, 2000) are directly injected into the soma to reflect an experimental *in vitro* condition. The firing pattern applied to the afferents will depend on the nature of the data available to tune the model. Preferably, simulations use the same stimuli as those applied experimentally, but it is common practice to consider each afferent as an independent renewal process, generating spikes with a Poisson or gamma interval distribution (chapter 7, section 2). As noted

earlier, correlated firing between the afferents can be needed to generate in the output neurons the variation observed in intervals between consecutive spikes (Harsch and Robinson, 2000; Fellous et al., 2003; Jolivet et al., 2006).

In an engineering context, sine waves and impulses are classically used for linear systems, whereas random broadband input signals or white noise are more suited for the identification of nonlinear systems (Marmarelis, 2004). Natural stimuli (often with $1/f$ spectral composition) are increasingly used and offer the advantage that the network is tested under its normal operating conditions.

We close this section with an example of a network component that is usually not modeled: extracellular space. The ionic composition of the extracellular space is commonly assumed to be spatially uniform and constant over time, which is reflected in the fixed values of the ionic Nernst potentials (equation 5.11). In resting conditions also, changes in the extracellular potential are considered too small to affect neighboring neurons' membrane potentials, currents, or conductances (Holt, 1998). Ehaptic field effects are nevertheless believed to contribute to the recruitment of neurons during very fast oscillations (Steriade, 2003) and, likewise, accumulation of ions in confined extracellular compartments during fast activity may change the Nernst potentials.

McIntyre et al. (2004) modeled extracellular space explicitly as a grid in order to solve the Laplacian equation describing the electric field around an electrode implanted for deep-brain stimulation. Some simulation packages, such as GENESIS and NEURON, provide tools for measuring the field potential at a particular location. Their use requires a multicompartmental neuron representation that allows axial currents to run along a dipole because the current balance in a single isopotential soma strictly precludes the generation of an extracellular electric field (see Gold et al., 2006).

13.4 Implementation, Validation, and Simulation of Large-Scale Network Models

At this stage, the major concern is to solve the system of differential equations that constitute the model. This is greatly facilitated by available simulation packages that offer script languages and a variety of solution engines. Most probably, the network also needs to be rescaled to fit the available memory and processing resources.

Choosing a Simulation Package

Historically, many of the first neuronal networks were solved with the electric-circuit simulation software SPICE (<http://www.topology.org/soft/sim.html>). Today there is a large diversity in software packages (see the software appendix for detailed information), as illustrated by their recent use: GENESIS (Maex and De Schutter, 2007), NEURON (Migliore et al., 2005a, 2006; Vida et al., 2006), SYNOD and its succes-

sor NEST (Diesmann et al., 1999), Matlab (Wilson et al., 2001; Häusler and Maass, 2007), XPPAUT (Marchetti et al., 2005), or SPLIT (Djurfeldt et al., 2005). See also Brette et al. (2007) for a more complete technical exposition and comparison of network simulation software.

GENESIS has a nice object-oriented structure with powerful connection commands, a flexible scripting interface, and a dedicated tool for fast numerical integration (the hsolve object). NEST has the benefit of transparent parallelization. The choice of simulation package may also be guided by the available neuron implementations (see <http://senselab.med.yale.edu/senselab/ModelDB>) so long as models remain specified by software-specific scripts that are not portable (see section 13.6).

Scaling the Network

The strategies most frequently used to bring the model to a feasible size are reduction of its spatial dimensions and dilution or rarefaction of the numbers of neurons and synapses. Instead of implementing the network in its full two- or three-dimensional extension, it may be worth starting with a one-dimensional version along the axis of primary interest, taking advantage of anisotropies in the actual circuit (as present for instance in the cerebellar cortex or hippocampus) or in the property being studied (such as orientation and directional selectivity in the visual cortex). Other strategies for reducing the actual three-dimensional network are the application of full-range connectivity (Häusler and Maass, 2007) or mapping the network on a ring (Vida et al., 2006), torus (Percha et al., 2005), or sphere (Izhikevich et al., 2004).

By offering circular boundary conditions, these latter techniques in addition avoid edge artifacts, an ever-present concern. In this respect, it is recommended that the weight of synapses be scaled in reverse proportion to their number, so as to keep the total synaptic current on each neuron or compartment approximately constant. This prevents too large heterogeneity in synaptic drive across the network, particularly at truncated boundaries and when connectivity is sparse. A more expensive method is to attach a boundary of void neurons, which are simulated like all other neurons but whose output is discarded from analysis (Wörgötter and Koch, 1991).

Rarefying the model network is almost unavoidable, even in studies modeling only a restricted patch of the actual circuit. The prime consequence of reducing the density of neurons again is the loss of synapses. It is possible to raise the firing rate of external (forward) afferents in order to correct for the reduction in their number, but lost internal (feedback) connections must be compensated for by an increase in the strength of the remaining synapses. Because the resulting enhanced variation of the postsynaptic potential may have a desynchronizing effect on the network (see section 13.3), it is also recommended that the connection density be adjusted when a network is rescaled, so as to keep the average number of synapses on each neuron constant.

Structural Validation of a Network

Structural validation of a network is the process of confirming that the constructed model matches the one intended in the design phase. For large networks, manual tracking is feasible for only a restricted set of parameters.

An obvious way to validate a model is visualization of its spatial organization, including cell positions, morphologies, and connections, which most simulators support (Gleeson et al., 2007; Cornelis and De Schutter, 2003). Automatic validation methods built into the scripting language gather statistics on single parameters or check the balance between two or more parameters. For instance, in networks with stochastically defined connections, each neuron can be checked to see whether it is receiving balanced numbers of synapses from different cell classes.

However, in large-scale networks, it is almost inevitable that some parameters remain unconstrained by experimental data, and it often requires a few test simulations to assign reasonable values to them (see also section 13.5). The use of indirect criteria may be necessary, such as the mean and variance of the firing rate across a neuron population for which the number, firing rate, or synaptic strength of the afferents is not known.

Numerical Considerations

Caution is needed in defining the initial conditions, in choosing an appropriate numerical integration method and associated time step, and in the proper use of random number generators (RNGs; see chapter 2, section 7). The choice of initial values for the model variables, and checking their consistency, is less critical when noise is added during the simulation. The dynamics of models lacking stochasticity of spike generation like continuous-valued firing-rate models, however, may be more sensitive to the initial conditions, particularly in networks with strong lateral inhibition. As a rule, control simulations must be run with different initial values of the seed variables of the RNGs, which should not qualitatively alter the results, and initial response transients must be discarded.

The computational size of large-scale model networks requires a numerical integration method that is stable and accurate for large time steps. The implicit Crank-Nicolson method has second-order accuracy and is always stable, but can induce damped error fluctuations for too large time steps (chapter 1, section 6). Our standard practice is to use 20- μ s integration steps, which in many models is the order of magnitude of the shortest time constant (that of activation of the fast sodium current). Controls should always be carried out, both using smaller time steps as well as alternative integration methods (such as an exponential Euler in 2- μ s steps). This is especially important when the conductances between compartments are large, as in networks with gap junctions. Also, networks of LIF neurons require special concern because artifactual synchronization has been reported for them (Hansel et al., 1998).

During a network simulation, the RNGs may be called upon an extraordinary number of times when stochastic inputs are applied. It is therefore of utmost importance to use RNGs with multiple seed variables, having sufficiently long periods of repetition (chapter 2, section 7; Tuckwell, 1995). Today simulators come with high-quality RNGs, but even then precautions are necessary. For instance, drawing the event times of a stationary Poisson point process from the associated Poisson interval distribution, instead of checking at every time step whether the Poisson point process should fire, dramatically reduces the number of RNG calls.

Recording Network Activity

Although most simulators have a graphical user interface, their heavy load precludes online monitoring of large networks. Most commonly, during the simulation spike times and time traces of the membrane potential or other variables of interest are saved on disk and analyzed offline with custom-written programs. Field potentials and EEG signals can be calculated with virtual electrodes like the efield object in GENESIS (M. Wilson and Bower, 1989). Alternatively, surrogates for these quantities can be calculated by summing together the membrane potentials (Traub et al., 1999b) or by compiling spike-time histograms from subpopulations of neurons (figure 13.4).

13.5 Analysis and Interpretation of Computed Network Activity

As stated earlier, the best way to avoid painstaking data analysis is to start from a clear scientific question, although admittedly many large-scale network models are exploratory in nature. All networks require validation in order to detect anomalies caused by errors in the software or the model's implementation, or by poor tuning of unconstrained parameters. Further analysis is directed toward comparing network activity with experimental data or, in its absence, making predictions for new experiments.

Functional Model Validation

For most network models, the output will be a large set of spike trains. Anomalies such as overexcitation (owing to too strong recurrent excitation) or a large fraction of neurons remaining silent (owing to too strong lateral inhibition) will be evident from simple inspection of the raster plots of spike times. It is also of utmost importance to view the raster plot of input spikes, which can show unexpected stripe patterns, owing to resonance in a poor quality RNG. Especially at low firing rates, the output of RNGs can deviate from an intended Poisson process.

A few neurons should have their recorded spike times carefully checked against their membrane potential traces. Spikes may be absent from the recording as a result

of depolarization block of the neurons, or they may have been missed when the threshold or refraction time for spike detection was set too high.

Spike Train Analysis

Spike train analysis is a discipline apart (for an introduction, see Gabbiani and Koch, 1998), and its application is not different for recorded or computed data, other than the experimentalist's spike-sorting problem. Note that many commercial tools available for experimental analysis can also be used for simulation data. We routinely apply the following procedures in both experiment and simulation (Vos et al., 1999).

Tests Performed on Individual Neurons

1. The mean neuronal spike rate, and its distribution across the population.
2. The distribution of interspike intervals (ISI histogram). Often, regularity of spiking is derived as the coefficient of variation (SD/mean), but the long tails of ISI histograms advocate the use of other measures like MAD/median (MAD: median absolute deviation), or short-term measures like CV_2 (the difference of two consecutive ISIs divided by their mean).
3. An additional test for regularity of firing compares the distribution of ISI_{i+1}/ISI_i with that of a Poisson spike train, which has a $1/(1+t)^2$ distribution.
4. The autocorrelation histogram (ACH), which, for point processes, gives the distribution of the interval between any pair of spikes (not just between consecutive spikes as in the ISI). It is most convenient to express the ACH values as spike rates (Vos et al., 1999), in which case the ACH displays the spike rate the neuron produces in windows preceding and following its own spikes, averaged over all spikes in the record.

Tests Performed on Neuron Pairs

5. The cross-correlation histogram (CCH) displays the average spike rate of the target neuron preceding and following spikes of the reference neuron. Note that features on the ACHs of target and reference neurons (caused by their intrinsic dynamics) can shape the right and left halves of the CCH, respectively (G. P. Moore et al., 1970). Common flaws in the use and interpretation of the CCH method are discussed later.

Test Performed on the Population Response

6. The spike-time histogram that monitors the network spike rate in discrete time bins throughout the simulation can sometimes be used as a surrogate measure of the local field potential (figure 13.4). From the autocorrelogram of this spike-time histogram or from its spectrogram, synchronous oscillations can then be quantified, although many other useful measures of synchrony have been proposed (Vida et al., 2006; Percha et al., 2005).

It is not always feasible to calculate and inspect pairwise CCHs in large datasets. Algorithms were developed to identify synchronously active clusters or assemblies, for instance, in model networks with stochastic or plastic connections (Izhikevich et al., 2004), or to extract distributed information by analyzing only a random sample of the available spike trains. An interesting trend in spike train analysis is to have a “readout” neuron do the job (Häusler and Maass, 2007). The underlying assumption is that a network involved in a particular task, such as the recognition, discrimination, or completion of input spike patterns, should generate output spike trains from which all the information needed for accomplishing that task can be easily extracted. This extraction is done by a downstream artificial neuron (the readout neuron), to which all output neurons feed and which is trained to perform the desired task.

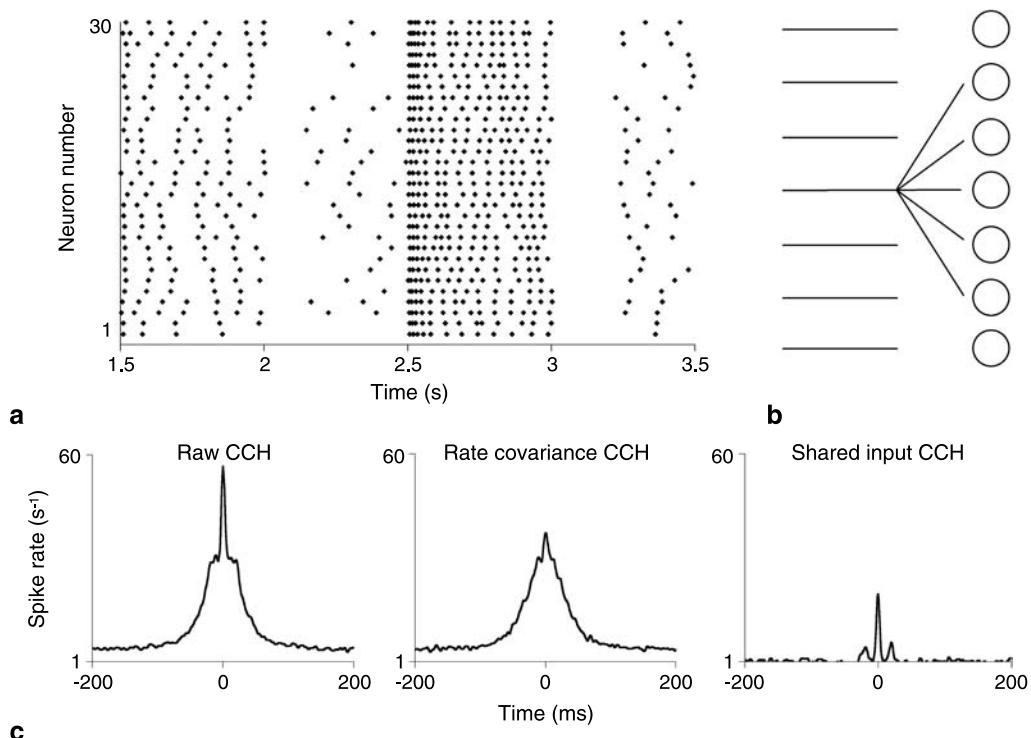
Comparison with Experimental Data

A difficult issue is the comparison of simulated and experimental data and, associated with it, the comparison of spike trains obtained at different levels of network excitation. We discuss some limitations on the use and interpretation of the most popular tool, the CCH.

To begin with, simulations are often run with stationary inputs, whereas there is no control about stationarity in experimental recordings. Trying to avoid the effects of nonstationarity by analyzing shorter time windows raises the problem of significance. These problems can be partly circumvented by applying appropriate normalization procedures. For instance, expressing the height of the central peak on the CCH, a measure of synchrony, as a Z-score (deviation from the mean in units of standard deviation) yields a value that in our experience is less dependent on the length of the recording than is the peak height or the average ratio of height to background (Vos et al., 1999).

When network activity at different firing rates is to be compared, the area of the central peak (the excess number of spikes fired by the target neuron around a spike of the reference neuron) offers a better measure of synchrony than peak height because peak width decreases with firing rate, whereas peak area remains almost constant (Maex et al., 2000). Hence measures of synchrony must be properly normalized when states of different firing rates are compared, and it is erroneous to conclude that the degree of synchrony has changed when only spike occurrences were counted within a fixed interval about the central peak. It is also worth noting that since both common excitation and common inhibition synchronize their targets, the presence of a central peak gives no indication about the sign of shared inputs.

A particular problem in deriving functional connectivity from experimental recordings is rate covariance, which reflects spurious synchrony of the target neurons caused by the simultaneous activation of their *non-common* afferents, as illustrated for a network simulation in figure 13.6. Similar interpretation problems arise for the

**Figure 13.6**

Rate covariance confounds the interpretation of the cross-correlation histogram. Simulations of a cerebellar feedforward network (see b) of 30 neurons (Golgi cells), each receiving excitation from 108 afferent parallel fibers (see Maex et al., 2000). Each Golgi cell shared 18 of its 108 afferents with its neighbor on either side. The Poisson processes generating afferent spikes had their rates simultaneously and randomly varied every 0.5 s, causing the entire neuron population to co-vary its firing rate as shown in the raster plot (a). The overall cross-correlogram (c, left) shows a sharp peak on top of a broader peak. These peaks can be dissected (in this case by filtering) and represent genuine synchrony caused by common synapses (c, right) and synchrony reflecting variations in the firing rate in noncommon afferents (c, middle), respectively.

network spectrogram, on which it is not uncommon to find a peak centered at the networks' average firing rate, even in the absence of synchronization.

Finally, neurons may synchronize in a manner undetectable with linear statistical measures like the CCH. Nonlinear methods are currently being developed to detect fleeting phase locking in multichannel recordings, and their reliability is tested in network models (Le Van Quien and Bragin, 2007; Pereda et al., 2005; Chávez et al., 2003).

Physiological Inference

As already stated, network models should be built with a particular question in mind, and overinterpretation of the data should be avoided. Common views regard-

ing the predictive power or physiological relevance of network models range from too optimistic to too pessimistic, as attested by the following two extreme attitudes.

“Proof of Existence Is Proof of Uniqueness”

This optimistic view ignores the fact that many network models use insufficiently constrained parameters, through which they keep a certain “pluripotency” and could be tuned, upon proper parameter assignment, to produce a diversity of behaviors. (This is not to say of course that actual networks cannot be multifunctional.) Hence the reproduction of a single set of experimental data does *not* suffice to validate a network model. The credibility of claims about relevance can be enhanced by testing the model’s sensitivity to variations in parameter values (robustness; chapter 12, section 1), by trying to simplify the model to its most essential parts (compactness), and by testing the model’s ability to reproduce experimental responses to stimuli other than those used for tuning its parameters (generalization).

The misconception described here also points to a limitation that most current network models still suffer from: they are useful for revealing potential mechanisms for generating a desired behavior (for instance, oscillations in an inhibitory network), but owing to a lack of biological detail, they fail to identify the actual neuron populations involved.

“Of Which Parameters Are You Sure?”

The pessimism expressed in this question is only partly justified. Unlike “wet” experiments, computer models offer a unique opportunity to vary and control parameters and to examine their effects on network dynamics. The predicted dynamics of a network helps formulate new hypotheses, which can be as useful for our understanding as the experimental measurement of a component parameter. Finally, one may wonder whether a perfect brain-copy model, with all its parameters specified, would add much to our understanding, apart from providing of course the ideal “experimental” subject.

13.6 Developments in Software Engineering for Network Modeling

Over the years, large-scale modeling projects have stimulated progress in network specification, parallel processing, and memory sharing. We discuss some of these developments in this section (see the software appendix for programs mentioned).

Model Specification

Model descriptions should be concise so as to be easily readable, but complete enough to allow an identical reinstatement of stochastically defined structures or parameters. Purely mathematical descriptions often suffer from errors in print, whereas program scripts are not portable to other simulation engines. Ideally, new

tools should allow transparent distributed interactive collaboration on the same model, benchmarking of the model against experimental data, and direct instantiation of the model from databases. The descriptions can take advantage of the hierarchical and modular design of a network. As stated earlier, network models are conceptually built by delineating neuron classes and applying (deterministic or stochastic) connection patterns to all instantiations of each representative neuron model.

Until the 1980s, network models were specified by their low-level implementation. The structure of the network model and its parameters were tightly linked to the program code and intertwined with the numerical algorithms and the commands controlling access to disks and other peripheral resources. Later implementations separated out the biological parameters and structures from the numerical algorithms. These were the first-generation general neural network simulators, and they could implement a range of network types. Nowadays, to accommodate the complexity of large network models, the better option is to use a neuronal simulator like NEURON, GENESIS, or NEST. These simulators separate numerical parameters like time step or accuracy, from modeling parameters like channel activation thresholds and time constants, and allow network models to be specified with a mathematical precision, so that given the same set of parameters, they instantiate the same model. This facilitates tracking the exact structure of the model and performing automated statistical validation, including calculation of the mean and standard deviation of randomized parameters like the numbers of connections. It is worth mentioning that it is always possible to identically realize a stochastic network by reinitializing the seed variables of the RNGs at least twice: before stochastically connecting the network and before starting the generation of random inputs.

Unfortunately, all current simulation software uses a programming paradigm to describe the biological components of a network model. This has several drawbacks. As already mentioned, the program code describing the model parameters is mixed with periphery code controlling the output or with statements for partitioning the model, making it difficult to extract relevant parameters from a large network model. Second, the program is implemented in a simulator-specific scripting language, so that porting the model to a different simulator involves recoding the model in the scripting language of the target simulator. Third, simulator-specific scripts impede automatically connecting simulators to databases.

Two approaches are currently being taken to address these problems. The first approach takes up some of the ideas behind the World Wide Web. The key is that a model should be described declaratively instead of being coded algorithmically if it is to be instantiated from a database. neuroConstruct (Gleeson et al., 2006) and Neurospaces (Cornelis and De Schutter, 2003) allow purely declarative network

specification. neuroConstruct uses neuroML to obtain simulator independence. The neuroML standard (Goddard et al., 2001) is specifically aimed at model portability by developing communication protocols between databases and simulators. Neuro-spaces builds a mathematical representation from the biological specification of the network model and is being developed in the context of the GENESIS-3 project (Cornelis and De Schutter, 2003; Beeman and Bower, 2005), along with the MOOSE simulation kernel.

The second approach adds a software layer on top of the core simulator, using a standard scripting language like Python. This new software layer uses libraries to translate a model specified in Python, using custom scripts, into simulator-specific code (Davison et al., 2009). Several simulators, such as TOPOGRAPHICA (Bednar et al., 2004), NEST, and NEURON, support this feature. One of the goals of portable network specification is the interoperability of different simulation engines on the same model, each optimized for one model component or process (such as diffusion, reaction kinetics, and electrotonic propagation) (Cannon et al., 2007).

Distributed Processing

Simulations of large network models may exceed the capacity of personal computers and workstations, hence computer clusters or supercomputers are needed. Not all simulators support parallelization, and if the cluster is shared among different researchers, clustering or grid software that controls and authorizes access to the parallel hardware may need to be installed.

Preparing the Model

The model must be partitioned into parts, each allocated to a different central processing unit (CPU). The partitioning process puts different equations of the model on different nodes under the assumption that they can be solved independently for a short simulation time. The goal is to balance the load of all CPUs, or to minimize the total number of idle CPU cycles. With static load balancing, the partitions are computed once and do not change during the simulation. With dynamic load balancing, the partitions can change over time, although this strategy has not yet been applied for biologically realistic network models.

Many simulators have a scripting language that provides explicit commands for nonautomated parallelization and partitioning of a model. The NEST and NEURON simulators provide solutions if the neuron models in the network are all of the same type (Migliore et al., 2006; Morrison et al., 2005). Solutions for automatic partitioning of highly heterogeneous network models are being developed (Cornelis and De Schutter, 2007).

Most partitioning methods slice the network model at cell boundaries (for a notable exception, see Hines et al., 2008a). The axonal delays allow the partitions to run

isolated from each other for a brief time. This technique was borrowed from distributed discrete-event simulation (Misra, 1986) and has been adapted for parallel neural network simulations (Migliore et al., 2006; Morrison et al., 2005). The key element of the technique is that a spike traveling along the axon is not modeled explicitly (section 13.3). Instead, the action potential is translated into a discrete event and put in a global or distributed queue. After the axonal delay elapses, the event is popped from the queue and translated into a function call at the target site, saving the latter from continually having to poll the queue.

Model Partition Tuning

Depending on the model in question, defining a suitable partition for each CPU can be either easy or difficult. For a heterogeneous cerebellar cortex model in pGENESIS (with nonautomated partitioning), the load of one Purkinje cell counterbalanced the load of 2,200 granule cells (Howell et al., 2000). The best way to find optimal partitions for such heterogeneous network models is to monitor the performance of the software by investigating the load of the parallel computer during small initial simulation runs, and to take the necessary corrective actions.

A search for the optimal load balance for a model is based on two different criteria. The first criterion takes account of the CPU load for each processor. For large-scale networks with a high degree of biological realism, the CPU load is often determined by the number of synapses, which easily exceeds the number of compartments and channels assigned to that CPU. If we assume interneuronal communication using discrete events as explained earlier, these network models are bounded by local CPU overhead and not by communication between CPUs (Djurfeldt et al., 2005). However, when dendritic gap junctions or graded transmitter release are modeled, communication overhead can start to have a large impact on overall performance.

The second criterion concerns interprocessor communication. As stated earlier, the communication overhead between the CPUs of a parallel computer for most networks is due to the connectivity by axons. Partitioning schemes that result in fewer connections between partitions, and longer connection delays, will help the simulator reach an optimal load balance. Obviously, there is a tradeoff between these criteria: A single partition results in maximal CPU overhead whereas the maximum amount of partitions, obtained by putting each cell in a separate partition, results in a maximal communication overhead. Benchmarking of simulation performance is needed to find an optimal balance between these two criteria.

In addition, some factors contributing to the complexity of distributed processing are specific to the software and hardware used. Indeed, the choice of algorithms and data structures has a dramatic impact on performance. The overall network connectivity can either be determined before parallel runtime (e.g., some NEURON simulations) or during runtime (e.g., pGENESIS). The first approach will greatly diminish

setup time of the parallel simulation but may run into memory overflow problems for complex networks because it requires duplication of the full network topology on all nodes. pGENESIS uses sophisticated algorithms to set up the network projections for parallel simulations. The advantage of these algorithms is that they take three-dimensional dendritic positions into account when defining the connections. They accommodate arbitrary network models but at the cost of internode communication during the model's construction phase. Depending on the exact partitions in the model and the connections between them, this can easily account for a third of the total application running time.

The hardware architecture of a parallel computer can also have an impact on simulator performance. Clusters of workstations mostly communicate via fast Ethernet-type connections. As a consequence, each node in the cluster can communicate directly with any of the other nodes. On the other hand, some supercomputers, including IBM's Blue Gene/L supercomputer recently used to optimize performance of the SPLIT simulator (Djurfeldt et al., 2005), consist of nodes interconnected in a mesh. For such computers, node-to-node communication can involve many hops instead of one. In addition, if the simulation output involves nonlocal disk access, it can interfere with the hardware architecture and also become a bottleneck.

Parallel Model Validation

Structural validation of parallel models is difficult since the model is physically partitioned over many processors. The importance of validation is twofold. First, the outcome can reveal problems with the software platform for that particular simulation, such as inappropriate scaling of the method used to partition the model or problems concerning the numerical methods, such as random number generation. Second, an unexpected outcome can indicate that upscaling induced an unforeseen structure on the network. The partitions on different nodes can be partially overlapping or can show holes not present in a serial version of the model. This kind of problem requires additional investigation on a per-model basis. Automated validation can prevent some of these problems.

Technical Optimizations

The low-level details of the implementation of a large network are unique for each simulator and unfortunately the user will need to study them closely. One example is the axonal connections, which each simulator implements in a different way. Nevertheless, some of these details can be exploited to optimize the implementation of a network model, as illustrated here by the sharing of function tables. Neuron model instances of the same population share a number of characteristics, such as the kinetics of their calcium- and voltage-dependent potassium currents. Putting the pre-computed kinetics in a table saves costly computation time. Moreover, if the table is shared among neurons of the same population, then huge memory savings are

achieved and in addition, a CPU can do faster lookups (according to the principle of locality of information). Both NEURON and GENESIS provide explicit commands to construct channel kinetic tables, and the GENESIS website provides a tutorial on how to construct a network model with shared function tables (<http://genesis-sim.org/GENESIS/gum-tutorials/cornelis>).

13.7 Some Notes on Neuronal Network Dynamics

Feedforward versus Feedback

We start from equations (13.1) and (13.2). If the synaptic input I_i^{network} depends in one way or another on V_i , then the circuit is of the feedback type; otherwise it is connected in a purely forward manner. Hence, connectivity described as reciprocal, mutual, recurrent, or lateral (the latter stressing the spatial extension of within-layer connections) implies feedback coupling, irrespective of the anatomical hierarchy or the absence thereof. Owing to their (primarily local) lateral connectivity, most actual networks appear to have feedback coupling. Nevertheless, their responses may contain a fast feedforward component, which is due to the delay it takes for feedback spikes to be fully recruited.

Some network models, such as the synfire chain, have only forward connections laid between successive layers (Abeles, 1991; Diesmann et al., 1999). With properly chosen patterns of spatial convergence and divergence, properties of feedback networks may be mimicked by repeated forward propagation through a chain of layers (Varjú, 1962).

Feedback Excitation versus Inhibition

In a simple firing-rate model, feedback excitation takes the form

$$\tau \frac{dV}{dt} = -V + RI + w[V]^+, \quad (13.7)$$

with τ denoting the time constant (equal to $R_m \times C_m$, the actual parameter determining dynamics); R , input resistance; w , connection strength or gain ($0 \leq w < 1$); and where $[V_i]^+ = V_i$ if $V_i > 0$, and vanishes otherwise. Assuming V to be positive and rewriting equation (13.7) in its canonical form,

$$\frac{\tau}{1-w} \frac{dV}{dt} = -V + \frac{RI}{1-w}. \quad (13.8)$$

It is clear that the effect of positive feedback is an enhancement of the response and a prolongation of the time constant, both by the factor $1/(1-w)$. Hence, feedback excitation operates as an amplifier, which becomes unstable at high gains.

In contrast, feedback inhibition attenuates responses by a factor $1/(1+w)$, and accordingly the steady state is approached faster. Note that as a consequence of the stability constraint ($w < 1$), linear self-inhibition often fails to reduce a neuron's output by more than 50% and must therefore be supplemented by other gain control mechanisms, such as feedforward inhibition or shunting inhibition (see section 13.3).

Reciprocal or lateral inhibition accentuates small differences in input, as shown here for a pair of neurons receiving the same constant input I except for a finite amount ΔI :

$$\begin{aligned}\tau \frac{dV_1}{dt} &= -V_1 - w[V_2]^+ + R(I + \Delta I) \\ \tau \frac{dV_2}{dt} &= -V_2 - w[V_1]^+ + RI.\end{aligned}\tag{13.9}$$

Algebraically solving for the steady state gives

$$\begin{aligned}V_1 &= R \frac{I(1-w) + \Delta I}{1 - w^2} \\ V_2 &= R \frac{I(1-w) - w\Delta I}{1 - w^2},\end{aligned}\tag{13.10}$$

which demonstrates that the difference in the output is larger than that present between the inputs by the same factor $1/(1-w)$:

$$\Delta V = V_1 - V_2 = \frac{R\Delta I}{1-w} \geq R\Delta I.\tag{13.11}$$

However, the latter equation holds only so long as both neurons show a positive output (since neurons are threshold devices). This requires for the neuron receiving the weaker input that

$$V_2 > 0 \Rightarrow \frac{\Delta I}{I} < \frac{1}{w} - 1.\tag{13.12}$$

If this inequality is violated, the neuron receiving the stronger input suppresses the weaker neuron to below threshold, so that the outputs become $V_1 = I$ and $V_2 = 0$. Such a so-called winner-takes-all effect can be misleading during network validation because an increased strength of inhibition w paradoxically enhances the firing rate of part of the network (V_1 in this example).

If we would add a third neuron reciprocally inhibiting the second neuron, then neuron two would lessen its inhibition of the first neuron, whose output will increase. The resulting disinhibition is the hallmark of lateral inhibition (Hartline and Ratliff, 1972).

Inhibition located close to the soma is important for the control of spike timing. At dendrites, it may limit the window of synaptic integration, particularly in configurations where monosynaptic excitation is followed by disynaptic inhibition.

Linearity

Linearity is an important characteristic in systems analysis. For a linear system, the superposition property holds: the response to a compound stimulus equals the sum of the responses to each component. The response to a single pulse completely characterizes a linear system. Since stimuli can be considered as (temporal or spatial) series of pulses or sine waves, the output of a linear system can always be computed from knowledge of the impulse response (or its Fourier transform, the transfer function). This linear-summation property, however, severely limits the processing capabilities of linear systems (Poggio and Reichardt, 1976; Marmarelis, 2004). Most tasks, such as discrimination or decision, require a nonlinear operation. This insight has led to a search for nonlinear operations in neural circuits (Koch, 1999).

The firing threshold is a hard nonlinearity, which can be smoothed by noise or spontaneous activity (see section 13.3). However, the mass action of networks can sometimes be described as almost linear (Freeman, 1975), which may be accomplished, for instance, through push-pull connectivity in which neurons excited by a particular stimulus equally receive inhibition from neurons responding to the complementary (negative) stimulus.

Another nonlinear component in network operation is shunting synapses (see section 13.3). The distinction between hyperpolarizing (linear) and shunting (divisive) synapses can be appreciated from the equation describing a passive membrane (note that voltages are taken relative to the leak or resting potential):

$$c \frac{dV}{dt} = -g_l V - g_e(V - V_e) - g_i(V - V_i), \quad (13.13)$$

where the l, e, and i subscripts refer to the leak, excitatory, and inhibitory conductances, respectively. From this, the steady-state voltage V_{ss} is

$$V_{ss} = \frac{g_e V_e + g_i V_i}{g_l + g_e + g_i}. \quad (13.14)$$

Hence a GABA_A receptor channel having a reversal potential V_i much more negative than the resting potential can, even for small conductance g_i , provoke a large hyperpolarizing current $g_i V_i$, of which the effect is almost purely subtractive because of the negligible effect of a small g_i on the denominator. In contrast, if the driving force V_i is close to zero, then large increases of g_i can still reduce V_{ss} by increasing the value of the denominator, the effect now being almost purely divisive because of the negligible effect on the numerator.

Time Scales

Before starting up simulations, it is essential to know the range of time scales operating in the network. The connection delays, the membrane time constant, and the rates of all other processes should be carefully assessed because they can vary over orders of magnitude (the H-current, for instance, may require several hundreds of milliseconds to fully activate). The shortest and longest time scales are particularly important as indicators of the numerical integration step and the minimum time course to be simulated, respectively.

Many parameters will affect network dynamics in an easily predictable manner. It is useful to consider neurons and synapses as filters with an expected effect in the temporal domain. Inclusion of short-term plasticity, for instance, will markedly prolong the memory content of the network. Short-term facilitation will accentuate slow signal variations, whereas processes like short-term depression or firing-rate adaptation accentuate fast transitions (Abbott and Regehr, 2004). Above all, as noted earlier, the network architecture itself can extend the time scale of operation up to the seconds range, either by feedback excitation or reciprocal inhibition (Cannon and Robinson, 1985).

Delayed feedback is the mechanism underlying most physiological oscillations (Friesen et al., 1993). The resonance frequency of inhibitory networks is about $1/(4d)$, with d the average conduction delay (Brunel et al., 2003; Maex and De Schutter, 2003; Ratliff et al., 1969). Reciprocal excitation induces oscillations of $1/d$ frequency.

Nonlinear systems can exhibit complex interactions among processes with different time courses, so that the effects of firing-rate adaptation or reciprocal inhibition on synchronization are not easily predictable (section 13.3). A unifying framework for the stability of synchrony was provided by Gerstner et al. (1996), who concluded that synchrony is asymptotically stable if the neurons produce their spikes only during the ascending (depolarizing) courses of their membrane potentials.

13.8 Recommended Reading

One of the first networks that was successfully investigated both experimentally and analytically was the *Limulus* retina, and the papers by Hartline and Ratliff remain classics in the field (Hartline, 1967; Hartline and Ratliff, 1972). Compartmental network modeling is largely based on work initiated by Rall (see Segev et al., 1994), and many of its standards were set with the GENESIS project at the Woods Hole summer schools in computational neuroscience (M. Wilson and Bower, 1989; Wörgötter and Koch, 1991). The work of Traub and colleagues (1999a) provides examples of fruitful long-term interactions between models and experiments. Finally, two papers

are mentioned for the insights they deliver into what biological networks can be expected to do. Considering the variation in response characteristics within cell classes, David Robinson (1992), a pioneer in the application of engineering concepts to systems neuroscience, warns those expecting that the brain has been wired by an engineer. Stewart (2004) gives an illuminating account of the dominant impact of network architecture on network dynamics.

Acknowledgments

Marc-Oliver Gewaltig, Sean Hille, and Astrid Prinz provided invaluable comments on drafts of the manuscript.

Software Appendix

Updated information on software packages can be found on the book website at <http://www.compneuro.org>

CalC: Calcium Calculator

CalC is a deterministic simulator of calcium diffusion for modeling intracellular calcium concentration dynamics in 3-D, 2-D or 1-D in Cartesian or curvilinear cylindrical and spherical coordinate systems. CalC simulates the calcium channel influx and subsequent calcium buffering and binding to calcium receptors. An arbitrary number of mobile or fixed calcium buffers may be included; only one-to-one calcium binding stoichiometry is currently implemented. In Cartesian coordinates, the diffusion volume is given by a union of rectangular boxes and may include rectangular obstacles representing vesicles or other diffusion barriers. The distinguishing feature of CalC is its computational efficiency, which is due to the use of an implicit second-order accurate finite difference scheme. CalC allows the user to view the simulation results in real time using the xmGrace package. CalC is driven by a user modeling script with flow control features, which distinguishes it from graphical user interface (GUI)-based tools.

Author:

Victor Matveev, Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ, USA.

Programming language: C++

License: open source

URL: www.calciumpcalculator.org

Compiled binaries: Windows (32 bit), cygwin on Windows (32 bit)

References:

Modeling scripts for these and other publications are available at www.calciumpcalculator.org.

Matveev V, Sherman A, Zucker RS (2002) New and corrected simulations of synaptic facilitation. *Biophysical Journal* 83: 1368–1373.

Matveev V, Zucker RS, Sherman A (2004) Facilitation through buffer saturation: Constraints on endogenous buffering properties. *Biophysical Journal* 86: 2691–2709.

Matveev V, Bertram R, Sherman A (2006) Residual bound Ca^{2+} can account for the effects of Ca^{2+} buffers on synaptic facilitation. *Journal of Neurophysiology* 96: 3389–3397.

Catacomb: Components and Tools for Accessible Computer Modeling in Biology

Catacomb is a component-based modeling system that allows graphic construction and exploration of models of biological systems. Version 2 included a range of hard-coded components for modeling synapses,

ion channels, cells, and networks, as well as simple interactions between a simulated rat and its environment. In version 3, the component definitions are independent of the system itself and can be created or extended by the user via a graphical component editor and code fragments to implement their behaviors. Both the component specifications and the models built from them are declarative and can be exported as largely system-independent eXtendable Markup Language (XML). Internally, execution of the model involves the generation of Java source code from the model specification, which is then compiled with an embedded compiler before the model is actually run. The initial component sets focus on simple cells and networks for implementing goal-directed navigation.

Authors:

Robert Cannon, Cantanout Limited, Edinburgh, UK

Michel E. Hasselmo, Center for Memory and Brain, Department of Psychology, and Program in Neuroscience, Boston University, Boston, MA, USA.

Language: Java

License: general public license (open source)

URL: www.catacomb.org

Compiled binaries: yes, platform independent

References:

Cannon RC, Hasselmo ME, Koene RA (2003) From biophysics to behavior: Catacomb2 and the design of biologically-plausible models for spatial navigation. *Neuroinformatics* 1: 3–42.

E-Cell System

E-Cell System (version 3) is a software platform for modeling, simulating and analyzing complex, heterogeneous, and multiscale biochemical reaction systems like the living cell. A special property of the E-Cell System is its simulation core, a multialgorithm simulator that can execute composite models driven by any number or type of algorithms simultaneously. In the same model, for example, stochastic algorithms can be used to simulate reaction kinetics that are subject to fluctuations, while deterministic algorithms, based on differential equations, can be used for others. E-Cell System offers very high flexibility for advanced users to build their own plug-in algorithms and routines either through Python scripting or C++ programming. Run-time simulation can also be manipulated with Python scripting. On the other hand, novice users can build models using the graphical model editor and employ numerous established algorithms that are bundled with the software. Plug-in tools are also provided for parameter estimation and mathematical analysis. Simulations demanding high computational resources, such as estimation of parameters, can utilize distributed computing technology through its multisession scheduling tool.

Authors:

E-Cell System version 3 was conceived by Koichi Takahashi and further developed by Satya Nanda, Vel Arjunan, Gabor Bereczki, Kazunari Kaizu, Tomoya Kitayama, Masahiro Sugimoto, and Takeshi Sakurada at the Institute for Advanced Biosciences, Keio University, Fujisawa, Japan.

Programming languages: C++ (core and plug-in routines), Python (visual interface, run-time scripting, and plug-in routines), XML (E-Cell Model File, Systems Biology Markup Language)

License: general public license (open source)

Compiled binaries: Fedora Core Linux, Microsoft Windows

URL: www.e-cell.org

References:

Dhar PK, Tomita M, Arjunan SNV (eds.) (2007) *E-Cell System: Basic Concepts and Applications*. Austin, TX: Landes Bioscience.

Takahashi K, Kaizu K, Hu B, Tomita M (2004) A multi-algorithm, multi-timescale method for cell simulation. *Bioinformatics* 20: 538–546.

Takahashi K, Kitayama T, Ito D, Tomita M (2005) Distributed cell biology simulations with E-Cell System. In TT Wee, P Arzberger and A Konagaya, eds., *Grid Computing in Life Science*: 20–31. Singapore: World Scientific Publishing.

GENESIS

GENESIS (the GEneral NEural SImulation System) is a general-purpose software platform that was developed to support the biologically realistic simulation of neural systems, ranging from subcellular components and biochemical reactions to complex models of single neurons, simulations of large networks, and systems-level models. Parallel GENESIS (PGENESIS) is customarily used for large network simulations involving tens of thousands of realistic cell models, or for parameter searches.

The object-oriented approach taken by GENESIS and its high-level simulation language allow modelers to easily extend the capabilities of the simulator and to exchange, modify, and reuse models or model components. The scripting language and the modules are powerful enough that often only a few lines of script are needed to specify a sophisticated simulation. Typically, one creates a new GENESIS simulation by modifying one of the many example and tutorial simulations that are provided with GENESIS or that are available from the GENESIS web site.

Authors:

The GENESIS Developers Group

James M. Bower (principal investigator), University of Texas San Antonio and University of Texas Health Science Center San Antonio, USA.

Programming language: GENESIS and its graphical front end, XODUS, are written in C and run under most varieties of Unix with the X Window System.

Compiled binaries: Linux, Intel, and PPC on Mac OS X, Windows/Cygwin

URL: www.genesis-sim.org/GENESIS/

References:

Beeman, D (2005) GENESIS Modeling Tutorial. Brains, Minds, and Media. 1: bmm220 (urn:nbn:de:0009-3-2206). (<http://www.brains-minds-media.org>)

Bower JM, Beeman D (1998) *The book of GENESIS: Exploring realistic neural models with the GEneral NEural SImulation System*. 2nd ed. New York: Springer-Verlag.

Kinetikit

Kinetikit is a graphical environment for building and running models of biochemical signaling networks. It provides facilities for compartmental models of chemical reaction systems using mass-action as well as stochastic chemical kinetics with four algorithms. It can also build one-dimensional reaction-diffusion models. Kinetikit provides a click-and-drag interface for creating and deleting reaction entities and connecting them into chemical networks. The interface also supports numerous other simulation utilities, including dose-response curve generation and model comparisons. Kinetikit is built on top of the GENESIS simulator, using a special library of objects to represent chemical entities such as molecular species and reactions. It can also be run in a GENESIS script-driven batch mode. Kinetikit models can be read by the MOOSE simulator, which provides additional features such as a conversion routine to the standard Systems Biology Markup Language (SBML), and fast numerical engines.

Author:

Upinder S. Bhalla, National Centre for Biological Sciences, Tata Institute of Fundamental Research, Bangalore, India.

Programming language: GENESIS scripting language and C (hybrid)

License: open source

Compiled binaries: yes, for GENESIS

URL: www.ncbs.res.in/index.php?option=com_content&task=view&id=304&Itemid=9

References:

Bhalla US (2002) Use of Kinetikit and GENESIS for modeling signaling pathways. *Methods in Enzymology* 345: 3–23.

Vaytaden SJ, Bhalla US (2004) Developing complex signaling models using GENESIS/Kinetikit. *Science STKE* p14.

MCell and DReAMM: Monte Carlo Cell and Design, Render, and Animate MCell Models

MCell uses 3-D Monte Carlo algorithms to simulate movements and reactions of molecules within and between cells—cellular microphysiology. Simulations are designed using Model Description Language (MDL) input files, and currently MCell3 supports simulation of diffusion and reaction in solution, on arbitrarily complex surface meshes representing membranes, and combinations thereof. MCell3's MDL includes a new syntax for reactions on surfaces and surface regions, as well as event counting. New runtime optimizations can substantially reduce computational costs (e.g., event scheduling, variable time and space steps).

DReAMM provides a graphical user interface to visualize MCell results and create conventional or stereo animations, including the locations and states of meshes and molecules in solution and on surfaces (up to thousands of mesh objects and regions and classes of molecules). DReAMM can also import and edit meshes from 3-D reconstruction and computer-aided-design software (e.g., blender.org), and export MDL files for MCell simulations.

Authors:

Boris Kaminsky, Jen-Chien Jack Chang, Markus Dittrich, and Joel R. Stiles, National Resource for Biomedical Supercomputing, Pittsburgh Supercomputing Center, Carnegie Mellon University, Pittsburgh, PA, USA.

Rex A. Kerr, HHMI Janelia Farm Research Campus, Ashburn, VA, USA.

Jed Wing and Thomas M. Bartol, Computational Neurobiology Laboratory, The Salk Institute, La Jolla, CA, USA.

Programming language: Model Description Language (MCell input files), C and C++

License: open source and National Resource for Biomedical Supercomputing (NRBSC) license (free to nonprofit academic users)

Compiled binaries: Linux and similar Unix systems, PC and Mac; various 64-bit architectures

URL: www.mcell.psc.edu

References:

Coggan JS, Bartol TM, Esquenazi E, Stiles JR, Lamont S, Martone ME, Berg DK, Ellisman MH, Sejnowski TJ (2005) Evidence for ectopic neurotransmission at a neuronal synapse. *Science* 309: 446–451.

Kerr RA, Bartol TM, Kaminsky B, Dittrich M, Chang J-C JC, Baden SB, Sejnowski TJ, Stiles JR (2007) Fast Monte Carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. *SIAM Journal on Scientific Computing* 30: 3126–3149.

Stiles JR, Bartol TM (2001) Monte Carlo methods for simulating realistic synaptic microphysiology using MCell. In E De Schutter, ed. *Computational Neuroscience: Realistic Modeling for Experimentalists*, pp. 1–23. Boca Raton, FL: CRC Press.

MesoRD

MesoRD is a tool for stochastic simulation of chemical reactions in spatially extended systems. In particular, it is an implementation of the Next Subvolume Method (NSM), which is an exact and efficient method for simulating the reaction-diffusion master equation. MesoRD relies on Systems Biology Markup Language for model input. MesoRD includes the option to perform deterministic simulations, facilitating direct comparison of deterministic and stochastic implementations for the same reaction-diffusion model. MesoRD is fast, owing to an efficient implementation of the NSM; it is flexible because the program is supplied with the SBML model definitions and runtime parameters from either a GUI (Windows) or command line flags (Unix); and finally, it is portable since it is implemented in C++.

Authors:

Johan Elf, David Fange, Johan Hattne, Uppsala University, Sweden

Programming language: C++

License: general public license mixed with BSD

Compiled binaries: Windows 32-bit

URL: mesord.sourceforge.net

References:

Elf J, Ehrenberg M (2004) Spontaneous separation of bistable biochemical systems into spatial domains of opposite phases. *Systems Biology* 1: 230–236.

Fange D, Elf J (2006) Noise-induced Min phenotypes in *E. coli*. *PLoS Computational Biology* 2: e80.

Hattne J, Fange D, Elf J (2005) Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics* 21: 2923–2924.

Modigliani

The Modigliani framework was developed for the stochastic simulation of neurons and networks thereof. It can model the many stochastic processes that are present within neurons and relate them to whole-cell behavior. Modigliani uses compartmental modeling to account for spatial interaction within cells. Each compartment can contain several stochastic processes (e.g., ion channel populations) described by Langevin-type ordinary differential equations, Markov processes, or specific random variables.

Detailed functionality was added for studying voltage-gated ion channels and whole-neuron behavior (it can bootstrap Markov and Langevin ion channel models from deterministic Hodgkin-Huxley-type conductance models or voltage-clamp data). However, Modigliani is generally able to simulate arbitrary stochastic reaction-diffusion-type biology (e.g., Ca^{2+} waves in muscles). Modigliani's implementation allows the user to dynamically switch among Langevin, Markov, and deterministic integrations while the simulation is running and choose the best method, given local stochasticity. Modigliani is written in ANSI C++ and uses object-oriented coding following biological terminology (e.g., plug-and-play “ion channel objects” into a “neuron membrane object” while the simulator runs). Modigliani was written for large-scale simulations: Langevin-type stochastic squid axons are simulated as fast as NEURON simulates the same deterministic axon. The user can interact with Modigliani using a simple, Matlab-syntax-compatible interpreter with OpenGL-based visualization or use a purpose-built compiled code adapted for computer cluster simulations (e.g., XGrid).

Author:

A. Aldo Faisal, Department of Engineering and Department of Zoology, University of Cambridge, Cambridge, UK.

Programming language: ANSI C++

License: own license; GPL-like (free) for academic use

Compiled binaries: Mac OSX 10.x, Linux (for cluster simulations).

Currently unsupported, but working Win 32 (cygwin), Solaris, OSF/1

URL: www.modigliani.co.uk

References:

Faisal AA, Laughlin SB (2007) Stochastic simulations on the reliability of action potential propagation in thin axons. *PLoS Computational Biology* 3: e79.

Faisal AA, White JA, Laughlin SB (2005) Ion-channel noise places limits on the miniaturization of the brain's wiring. *Current Biology* 15: 1143–1149.

MOOSE: Multiscale Object-Oriented Simulation Environment

MOOSE is the base and numerical core for large, detailed simulations, including computational neuroscience and systems biology. MOOSE spans the range from single molecules to subcellular networks, from single cells to neuronal networks, and to still larger systems. It is backward compatible with GENESIS and forward compatible with Python. A key goal of MOOSE is to embrace many simulation and numerical engines that span a range of levels of simulation and to provide the infrastructure to enable these to interoperate seamlessly. Currently MOOSE supports three main solvers: for neuronal biophysics using custom code, for deterministic reaction-diffusion simulations using the GNU Scientific Library

routines, and for three-dimensional Monte Carlo reaction-diffusion models using the Smoldyn engine. The MOOSE messaging system provides the infrastructure for parallel model building without requiring the solution engines or the user-defined classes to write parallel code.

Authors:

Upinder S. Bhalla, Niraj Dudani, Subhasis Ray, and Raamesh Deshpande, National Centre for Biological Sciences, Bangalore, India.

Language: C++

License: Lesser general public license (open source)

Compiled binaries: Linux (32 and 64-bit), Solaris, Windows (32 bit), MacOS Darwin

URL: moose.ncbs.res.in, moose.sourceforge.net

NCS: NeoCortical Simulator

The NCS is intended to accurately model large-scale, horizontally dispersed and vertically layered distributions of neurons characteristic of the mammalian neocortex. It compiles and runs on any server running Linux and MPI. Although user-specified active spike templates are usually used for large simulations, Hodgkin-Huxley channel equations can be specified for the rapid sodium and delayed-rectifier spike behavior; slow ionic membrane channel families are standard. Multicompartment neurons are connected by forward and reverse conductances without detailed cable equations. Synapses are conductance based, with depression, facilitation, augmentation, and spike-timing dependent plasticity. What makes the NCS unique is that it is optimized to run on large clusters, with very efficient synaptic message passing, enabling the simulation of networks of millions of cells and hundreds of millions of synapses. Also, the NCS can be run in either batch mode (ASCII setting file), or in real time for robotic control (via an included web port server application).

Authors:

Philip Goodman and Frederick Harris, Jr., Departments of Medicine and Computer Science, University of Nevada, Reno, USA.

Programming language: C++, with optional Python control scripts

License: open source

Compiled binaries: available for Linux

URL: brain.unr.edu

References:

Brett R et al (2007) Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience* 23: 349–398.

Macera JC, Goodman PH, Harris FC, Drewes R, Maciokas JB (2004) Remote-neocortex control of robotic search and threat identification. *Robotic Autonomous Systems* 46: 97–110.

Maciokas JB, Goodman P, Kenyon J, Toledo-Rodriguez M, Markram H (2005) Accurate dynamical models of interneuronal GABAergic channel physiologies. *Neurocomputing* 65: 5–14.

NEST

NEST is a tool for simulating large heterogeneous networks of point neurons or neurons with a small number of compartments. It is best suited for models that focus on the dynamics, size, and structure of neural systems rather than on the detailed morphological and biophysical properties of individual neurons. NEST is optimized for networks of spiking neurons and can represent spikes in continuous time. One network can contain different types of model neurons. Different synapse models implement synaptic mechanisms, such as short-term or spike-timing-dependent plasticity. A network can be structured into layers, areas, and subnetworks. Special nodes provide input or obtain measurements from the network. With version 2.0, NEST introduces a Python-based user interface (PyNEST) that allows users to define, run, and analyze

their models from within a single environment. NEST can take advantage of multiprocessor (multicore) computers and computer clusters to increase the available memory or to speed up the simulation. The tool is the reference implementation of the NEST Initiative, a contract-based collaboration among several academic and industrial research institutes.

Authors:

Marc-Oliver Gewaltig, Honda Research Institute Europe GmbH, Offenbach, Germany.

Markus Diesmann, RIKEN Brain Science Institute, Wako City, Japan.

The current developer team.

Programming language: C++, Python, scripting language

License: open source

Compiled binaries: no

URL: www.nest-initiative.org

References:

Eppler JM, Helias M, Muller E, Diesmann M, Gewaltig M-O (2008) PyNEST: A convenient interface to the NEST simulator. *Frontiers in Computational Neuroscience* 2: 12.

Morrison A, Mehring C, Geisel T, Aertsen A, Diesmann M (2005) Advancing the boundaries of high-connectivity network simulation with distributed computing. *Neural Computation* 17: 1776–1801.

Morrison A, Straube S, Plesser HE, Diesmann M (2007) Exact subthreshold integration with continuous spike times in discrete-time neural network simulations. *Neural Computation* 19: 47–79.

NSL: Neural Simulation Language

NSL is a simulation environment for modular brain modeling that addresses the needs of a wide range of users. For novice users interested only in an introduction to neural networks, NSL provides user-friendly interfaces and a set of predefined artificial and biological neural models. For more advanced users well acquainted with neural modeling, NSL offers more sophistication through extended visualization tools and programming. It may easily be linked to other software by doing direct programming in C++, Java, or Matlab, such as linking it to numerical libraries or robotic systems.

Authors:

Alfredo Weitzenfeld, Instituto Tecnológico Autónomo de Mexico, Mexico City, Mexico.

Michael Arbib, Brain Simulation Laboratory, University of Southern California, Los Angeles, USA.

Programming language: C++, Java, Matlab

License: open source

Compiled binaries: PC and Linux

URL: www.neuralsimulationlanguage.org, nsl.usc.edu

References:

Weitzenfeld A, Arbib MA, Alexander A (2002) *The Neural Simulation Language NSL: A System for Brain Modeling*. Cambridge, MA: MIT Press.

neuroConstruct

neuroConstruct is a software tool that facilitates the construction, visualization, and analysis of conductance-based neuronal network models in 3-D space. Simulations are controlled through a user-friendly GUI, and script files can be automatically generated for the NEURON and GENESIS simulation environments. Models are represented in a simulator-independent way, based on the NeuroML standards, which facilitates the construction, testing, and interchange of models by experimentalists and theoreticians. neuroConstruct is written in Java, allowing use on multiple operating systems. A number of published neuronal models from various brain areas, including the cerebellum, dentate gyrus, and cortex, have been recreated in neuroConstruct and are distributed with the application.

Authors:

Padraig Gleeson, Volker Steuber, and R. Angus Silver, Department of Physiology, University College London, UK.

Programming language: Java, XML

License: general public license

Compiled binaries: available for Mac, Windows, Linux

URL: www.neuroConstruct.org

References:

Gleeson P, Steuber V, Silver RA (2007) neuroConstruct: A tool for modeling networks of neurons in 3D space. *Neuron* 54: 219–235.

Neurofitter

Neurofitter is a software package that automatically searches optimal values for parameters of neuronal models. These parameters can be maximal conductances, kinetic parameters related to ion channels, synaptic strengths, or any other parameters of a neuron or network model. Neurofitter uses electrophysiological time series data recorded from neurons and compares them with the output traces generated by the models. What makes Neurofitter unique is that it uses the phase-plane trajectory technique to compare these traces. This technique makes the algorithm robust to time shifts in the data traces. Neurofitter uses general global optimization techniques such as evolutionary strategies and particle swarm optimization to find the best set of parameter values for the model.

Authors:

Werner Van Geit, Pablo Achard, and Erik De Schutter, Laboratory for Theoretical Neurobiology, University of Antwerp, Belgium, and Computational Neuroscience Unit, Okinawa Institute of Science and Technology, Japan.

Programming language: C++ The model with parameters to tune can be written in any neuronal simulator, such as GENESIS, NEURON, or NEST

License: general public license

Compiled binaries: available for Intel and PPC on Mac OS X, Windows/Cygwin, Linux Slackware

URL: neurofitter.sourceforge.net

References:

Achard P, De Schutter E (2006) Complex parameter landscape for a complex neuron model. *PLoS Computational Biology* 2: e94.

Van Geit W, Achard P, De Schutter E (2007) Neurofitter: A parameter tuning package for a wide range of electrophysiological neuron models. *Frontiers in Neuroinformatics* 1: 1.

NEURON

NEURON simulates individual neurons and networks of neurons that may have complex morphology and biophysical membrane properties. NEURON is designed around the notion of continuous cable “sections” that are connected to form any kind of branched cable and that are endowed with properties that vary continuously with position along the section. The goal is to keep the physical properties of the neuron entirely separate from the numerical issue of size of spatial compartments. A model is constructed using GUI tools or by programming with a built-in interpreter. Individual cells and networks, including gap junctions, may be simulated on a parallel computer or workstation cluster and will give quantitatively identical results regardless of cell distribution or number of hosts. NEURON may be used as the controller for a dynamic clamp under RTAI Linux and with a National Instruments M series DAQ. Even when the model is a differential algebraic system, i.e., involving extracellular fields and linear circuits, variable step integration methods can be used.

Author:

Michael Hines, Department of Computer Science, Yale University, New Haven, CT, USA (principal author).

Programming language: C, C++, user programs in Python and/or HOC

License: open source; makes use of general public license packages

Compiled binaries: Linux, Mac OS X, Microsoft Windows

URL: www.neuron.yale.edu

References:

Carnevale NT, Hines ML (2006) *The NEURON Book*. Cambridge, UK: Cambridge University Press.
Hines ML, Carnevale NT (1997) The NEURON simulation environment. *Neural Computation* 9: 1179–1209.

Migliore M, Cannia C, Lytton WW, Markram H, Hines ML (2006) Parallel network simulations with NEURON. *Journal of Computational Neuroscience* 21: 119–129.

PSICS: Parallel Stochastic Ion Channel Simulator

PSICS is a single-neuron simulator that includes efficient population-based methods for treating ion channels stochastically as well as deterministically. Channels can be expressed as kinetic schemes or in the Hodgkin-Huxley formalism and are given exact positions on the cell prior to spatial discretization so that exactly the same physical model can be computed at different resolutions. Models are defined purely declaratively using eXtensible Markup Language (XML) elements and are extensively preprocessed in Java, including the tabulation of voltage-dependent quantities. This yields a form suited for the core calculation, which is implemented in Fortran and can be run on machines without Java. On a GridEngine system, PSICS can submit separate runs, either separate stochastic instances or with different parameter values, via standard queues and can reassemble and merge the results on completion of the last calculation. A companion program, ICING (Interactive Channel INsertion GUI), provides a user interface for part of the model specification process and supports labeling morphologies, the specification and display of channel allocation rules, and 3-D visualization of cells and channel densities.

Authors:

Robert C. Cannon, Textensor Limited, Edinburgh, UK.

Matthew F. Nolan, Centre for Cognitive and Neural Systems, University of Edinburgh, UK.

Programming language: Java and Fortran

License: open source

Compiled binaries: requires a separate Java virtual machine, but includes embedded binaries of the Fortran components for MacOS X (Intel), Linux, and Windows

URL: www.psics.org

SNNAP: Simulator for Neural Networks and Action Potentials

SNNAP is a versatile and user-friendly tool for rapidly developing and simulating realistic models of single neurons and neural networks. It can model many features of neuronal function, such as ionic currents and their modulation by intracellular ions and second messengers; synaptic transmission; and hetero- and homosynaptic plasticity. Voltage-dependent conductances are described by Hodgkin-Huxley-type equations, and time-dependent synaptic conductances are described by second-order differential equations. Parameters, initial conditions, and output options are passed to the simulator through modular ASCII files. The modular design facilitates incorporating existing files into new simulations and allows users to develop libraries of files describing distinctive cells and neural networks. Users can develop models and run simulations through a graphical user interface with no need for programming. The user base of SNNAP includes educators who employ simulations to teach the principles of neuroscience and researchers who model experimental results and carry out computer simulations.

Authors:

Douglas A. Baxter and John H. Byrne, Department of Neurobiology and Anatomy, University of Texas Medical School at Houston, TX, USA.

Programming Language: Java

License: general public license; source code is available upon request

Compiled binaries: distributed as Java byte code

URL: snnap.uth.tmc.edu

References:

Av-Ron E, Byrne JH, Baxter DA (2006) Teaching basic principles of neuroscience with computer simulations. *Journal of Undergraduate Neuroscience Education* A40–A52.

Baxter DA, Byrne JH (2007) Simulator for neural networks and action potentials. *Methods in Molecular Biology* 401: 127–154.

Ziv I, Baxter DA, Byrne JH (1994) Simulator for neural networks and action potentials: Description and application. *Journal of Neurophysiology* 71: 294–308.

STEPS: STochastic Engine for Pathway Simulation

STEPS is a reaction kinetics simulator developed to model molecular reactions in detailed neuronal geometries. Its core simulation algorithm is Gillespie's SSA, modified to handle reaction-diffusion systems. Molecules are able to diffuse in three-dimensional tetrahedral discretizations that can represent complex geometry spanning a wide range of scales. It provides support not only for membrane-bound reactions and transmembrane fluxes, but also for reactions between soluble ligands. STEPS can also perform simulations in well-mixed conditions (i.e., without 3-D geometry or diffusion). The design and simulation of models is scriptable with Python. This includes specifying the reaction rules and diffusion constants, defining the cellular morphology and generating a mesh, running a simulation, manipulating its internal state, and generating output. The computational routines, however, are written in C++, eliminating overhead that might result from scripting.

Authors:

Stefan Wils and Erik De Schutter, Laboratory for Theoretical Neurobiology, University of Antwerp, Belgium, and Computational Neuroscience Unit, Okinawa Institute of Science and Technology, Japan.

Programming language: Python, C++ (hybrid)

License: general public license

Compiled binaries: n.a.

URL: sourceforge.net/projects/steps/

StochSim

StochSim is a stochastic simulator for (bio)chemical reactions. The particles are represented as individual software objects that react according to probabilities derived from concentrations and rate constants. Particles can have multiple states, which are encoded by vectors of state variables, therefore avoiding the combinatorial explosion. StochSim permits the user to create simple one- and two-dimensional spatial structures in which nearest-neighbor interactions of molecules can be simulated.

Authors:

Carl Firth, unaffiliated.

Tom Shimizu, Harvard University, Cambridge, MA, USA.

Nicolas Le Novere and Anika Oellrich, European Bioinformatics Institute, Cambridge, UK.

Thierry Emonet, University of Chicago, Chicago, IL, USA.

Michael North, Argonne National Laboratory, Argonne, IL, USA.

Programming language: C++, Python, Perl

License: general public license

URLs: stochsim.sourceforge.net

References:

Le Novère N, Shimizu TS (2001) STOCHSIM: Modelling of stochastic biomolecular processes. *Bioinformatics* 17: 575–576.

Morton-Firth CJ, Bray D (1998) Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology* 192: 117–128.

VCELL—The Virtual Cell

The Virtual Cell is a general software environment for modeling cell biological processes that is deployed as a free distributed application over the internet. Using a graphical user interface, kinetic, transport, and electrophysiological data describing individual reactions are associated with 2-D or 3-D experimental microscopic images that describe their subcellular locations. From the resultant mechanistic model, the software automatically generates reaction-diffusion-advection partial differential equations. These equations are then solved numerically to produce simulations of the biological process. Nonspatial models may also be simulated either deterministically or stochastically. The models are stored in a database that permits users to share the models with individual collaborators or to make them completely public. The database also permits searching for model components either from other VCell models or from public pathway databases. More than a thousand independent users worldwide have created models and run simulations with the Virtual Cell.

Authors:

James C. Schaff, Boris M. Slepchenko, Ion I. Moraru, and Leslie M. Loew, National Resource for Cell Analysis and Modeling, University of Connecticut Health Center, Farmington, CT, USA.

Programming language: Java, C++

License: LGPL, open source (in process of being released)

Compiled binaries: not available

URL: vcell.org, sourceforge.net/projects/vcell/

References:

Moraru II, Schaff JC, Loew LM (2006) Think simulation—think experiment: The Virtual Cell paradigm. In LF Perrone, FP Wieland, J Liu, BG Lawson, DM Nicol, and RM Fujimoto, eds., 2006 *Winter Simulation Conference*, pp. 1713–1719. Monterey, CA: World Scientific Publishing.

Schaff J, Fink CC, Slepchenko B, Carson JH, Loew LM (1997) A general computational framework for modeling cellular structure and function. *Biophysical Journal* 73: 1135–1146.

Slepchenko BM, Schaff JC, Macara I, Loew LM (2003) Quantitative cell biology with the Virtual Cell. *Trends in Cellular Biology* 13: 570–576.

XPPAUT

XPPAUT is a general numerical tool for simulating, animating, and analyzing dynamic systems. These can range from discrete finite-state models (McCulloch-Pitts) to stochastic Markov models, to discretization of partial differential and integrodifferential equations. XPPAUT was not specifically developed for neural simulations, but because of its ability to provide a complete numerical analysis of the dependence of solutions on parameters (“bifurcation diagrams”), it is widely used by the community of computational and theoretical neuroscientists. There are numerous online tutorials, many of which are geared to neuroscience. While it can be used for modest-sized networks, it is not specifically designed for this purpose and owing to its history, there are limits on the size of problems that can be solved (about 2,000 differential equations is the current limit). Rather than a pure simulator, XPPAUT is a tool for understanding the equations and the results of simulating the equations. XPPAUT uses a highly optimized parser to produce a pseudocode that is interpreted and runs very fast—at about half the speed of a directly compiled code. Since no compiler is required, XPPAUT is a stand-alone program and runs on all platforms that have an X-windows

interface available (Unix, MAC OSX, Windows, etc.). The program is open source and available as source and various binary versions.

Author:

Bard Ermentrout, Department of Mathematics, University of Pittsburgh, Pittsburgh, PA, USA.

Programming language: C

License: general public license

Compiled binaries: available for several platforms, including Linux, MacOS, and Windows

URL: www.math.pitt.edu/~bard/xpp/xpp.html

References:

Bard Ermentrout (2002) *Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students*. Philadelphia, PA: Society for Industrial Mathematics.

References

- Abbott LF, Gerstner W (2004) Homeostasis and learning through spike-timing dependent plasticity. <http://infoscience.epfl.ch/record/114304/files/Abbott04.pdf>
- Abbott LF, Nelson SB (2000) Synaptic plasticity: Taming the beast. *Nature Neuroscience* 3 Suppl.: 1178–1183.
- Abbott LF, Regehr WG (2004) Synaptic computation. *Nature* 431: 796–803.
- Abbott LF, van Vreeswijk C (1993) Asynchronous states in a network of pulse-coupled oscillators. *Physical Review E* 48: 1483–1490.
- Abbott LF, Varela JA, Sen K, Nelson SB (1997) Synaptic depression and cortical gain control. *Science* 275: 220–224.
- Abeles M (1991) *Corticonics: Neural Circuits of the Cerebral Cortex*. New York: Cambridge University Press.
- Abramowitz M, Stegun IA (1970) *Tables of Mathematical Functions*. New York: Dover.
- Achard P, De Schutter E (2006) Complex parameter landscape for a complex neuron model. *PLoS Computational Biology* 2: e94.
- Achard P, De Schutter E (2008) Calcium, synaptic plasticity and intrinsic homeostasis in Purkinje neuron models. *Frontiers in Computational Neuroscience* 2: 8.
- Agmon-Snir H, Segev I (1993) Signal delay and input synchronization in passive dendritic structures. *Journal of Neurophysiology* 70: 2066–2085.
- Aharon S, Parnas H, Parnas I (1994) The magnitude and significance of Ca^{2+} domains for release of neurotransmitter. *Bulletin of Mathematical Biology* 56: 1095–1119.
- Akay T, Haehn S, Schmitz J, Buschges A (2004) Signals from load sensors underlie interjoint coordination during stepping movements of the stick insect leg. *Journal of Neurophysiology* 92: 42–51.
- Akemann W, Knöpfel T (2006) Interaction of Kv3 potassium channels and resurgent sodium current influences the rate of spontaneous firing of Purkinje neurons. *Journal of Neuroscience* 26: 4602–4612.
- Albritton ML, Meyer T, Stryer L (1992) Range of messenger action of calcium ion and inositol 1,4,5-triphosphate. *Science* 258: 1812–1815.
- Aldrich RW (1981) Inactivation of voltage-gated delayed potassium currents in molluscan neurons. *Biophysical Journal* 36: 519–532.
- Aldrich RW, Corey DP, Stevens CF (1983) A reinterpretation of mammalian sodium channel gating based on single channel recording. *Nature* 306: 436–441.
- Alle H, Geiger JR (2006) Combined analog and action potential coding in hippocampal mossy fibers. *Science* 311: 1290–1293.
- Almers W (1978) Gating currents and charge movements in excitable membranes. *Reviews of Physiology Biochemistry and Pharmacology* 82: 96–190.
- Ambros-Ingerson J, Holmes WR (2005) Analysis and comparison of morphological reconstructions of hippocampal field CA1 pyramidal cells. *Hippocampus* 15: 302–315.

- Ambros-Ingerson J, Grover LM, Holmes WR (2008) A classification method to distinguish cell-specific responses elicited by current pulses in hippocampal CA1 pyramidal cells. *Neural Computation* 20: 1512–1536.
- Andersen O, Koeppen RE II (1992) Molecular determinants of channel function. *Physiological Reviews* 72: S89–S158.
- Andrews SS, Bray D (2004) Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Physical Biology* 1: 137–151.
- Angstadt JD, Calabrese RL (1991) Calcium currents and graded synaptic transmission between heart interneurons of the leech. *Journal of Neuroscience* 11: 746–759.
- Aradi I, Maccaferri G (2004) Cell type-specific synaptic dynamics of synchronized bursting in the juvenile CA3 rat hippocampus. *Journal of Neuroscience* 24: 9681–9692.
- Aradi I, Soltesz I (2002) Modulation of network behavior by changes in variance in interneuronal properties. *Journal of Physiology* 538: 227–251.
- Armstrong CM (1969) Inactivation of the potassium conductance and related phenomena caused by quaternary ammonium ion injection in squid axons. *Journal of General Physiology* 54: 553–575.
- Armstrong CM (1981) Sodium channels and gating currents. *Physiological Reviews* 62: 644–683.
- Armstrong CM, Hille B (1998) Voltage-gated ion channels and electrical excitability. *Neuron* 20: 371–380.
- Arsiero M, Lüscher HR, Lundstrom BN, Giugliano M (2007) The impact of input fluctuations on the frequency-current relationships of layer 5 pyramidal neurons in the rat medial prefrontal cortex. *Journal of Neuroscience* 27: 3274–3284.
- Ascoli GA (1999) Progress and prospectives in computational neuroanatomy. *Anatomical Record* 257: 195–207.
- Ascoli GA (2002) Neuroanatomical algorithms for dendritic modelling. *Network: Computation in Neural Systems* 13: 247–260.
- Ascoli GA (2006) Mobilizing the base of neuroscience data: the case of neuronal morphologies. *Nature Reviews Neuroscience* 7: 318–324.
- Ascoli GA, Krichmar JL (2000) L-Neuron: a modelling tool for the efficient generation and parsimonious description of dendritic morphology. *Neurocomputing* 32: 1003–1011.
- Ascoli GA, Krichmar JL, Nasuto S, Senft SL (2001a) Generation, description and storage of dendritic morphology data. *Philosophical Transactions of the Royal Society, Series B* 356: 1131–1145.
- Ascoli GA, Krichmar JL, Nasuto S, Senft SL (2001b) Computer generation and quantitative morphometric analysis of virtual neurons. *Anatomy and Embryology* 204: 283–301.
- Attwell D, Gibb A (2005). Neuroenergetics and the kinetic design of excitatory synapses. *Nature Reviews Neuroscience* 6: 841–849.
- Augustine GJ, Santamaría F, Tanaka K (2003) Local calcium signaling in neurons. *Neuron* 40: 331–346.
- Bacon JP, Altman JS (1977) A silver intensification method for cobalt-filled neurons in whole-mount preparations. *Brain Research* 138: 359–363.
- Badel L, Lefort S, Brette R, Petersen CC, Gerstner W, Richardson MJ (2008) Dynamic I–V curves are reliable predictors of naturalistic pyramidal-neuron voltage traces. *Journal of Neurophysiology* 99: 656–666.
- Badoiu M, Rudolph M, Piwkowska Z, Destexhe A, Bal T (2005) High discharge variability in neurons driven by current noise. *Neurocomputing* 65: 493–498.
- Badoiu M, Zou Q, Davison AP, Rudolph M, Bal T, Frégnac Y, Destexhe A (2006) Biophysical and phenomenological models of multiple spike interactions in spike-timing dependent plasticity. *International Journal of Neural Systems* 16: 79–97.
- Bailor W (2003) *Writing ImageJ Plugins-A Tutorial*, Upper Austria University of Applied Sciences, Hagenberg, Austria.
- Bardo S, Cavazzini MG, Emptage N (2006) The role of the endoplasmic reticulum Ca^{2+} store in the plasticity of central neurons. *Trends in Pharmacological Sciences* 27: 78–84.

- Barrett JN, Crill WE (1974) Specific membrane properties of cat motoneurons. *Journal of Physiology* 239: 301–324.
- Bartolozzi C, Indiveri G (2007) Synaptic dynamics in analog VLSI. *Neural Computation* 19: 2581–2603.
- Bartos M, Manor Y, Nadim F, Marder E, Nusbaum MP (1999) Coordination of fast and slow rhythmic neuronal circuits. *Journal of Neuroscience* 19: 6650–6660.
- Bartos M, Vida I, Frotscher M, Geiger JR, Jonas P (2001) Rapid signaling at inhibitory synapses in a dentate gyrus interneuron network. *Journal of Neuroscience* 21: 2687–2698.
- Beard NA, Laver DR, Dulhunty AF (2004) Calsequestrin and the calcium release channel of skeletal and cardiac muscle. *Progress in Biophysics and Molecular Biology* 85: 33–69.
- Bednar JA, Kelkar A, Miikkulainen R (2004) Scaling self-organizing maps to model large cortical networks. *Neuroinformatics* 2: 275–302.
- Beeman D, Bower JM (2005) Genesis 3 development plans. <http://www.genesis-sim.org/GENESIS/newgrant/G3/index.html>.
- Bekkers JM, Häusser M (2007) Targeted dendrotomy reveals active and passive contributions of the dendritic tree to synaptic integration and neuronal output. *Proceedings of the National Academy of Sciences of the USA* 104: 11447–11452.
- Bekkers JM, Stevens CF (1996) Cable properties of cultured hippocampal neurons determined from sucrose-evoked miniature EPSCs. *Journal of Neurophysiology* 75: 1250–1255.
- Bem T, Le Feuvre Y, Rinzel J, Meyrand P (2005) Electrical coupling induces bistability of rhythms in networks of inhibitory spiking neurons. *European Journal of Neuroscience* 22: 2661–2668.
- Ben-Ari Y (2002) Excitatory actions of gaba during development: The nature of the nurture. *Nature Reviews Neuroscience* 3: 728–739.
- Benda J, Herz AVM (2003) A universal model for spike-frequency adaptation. *Neural Computation* 15: 2523–2564.
- Bennett MR, Farnell L, Gibson WG (2000) The probability of quantal secretion near a single calcium channel of an active zone. *Biophysical Journal* 78: 2201–2221.
- Benz R, Frohlich O, Lauger P, Montal M (1975) Electrical capacity of black lipid films and of lipid bilayers made from monolayers. *Biochimica Biophysica Acta* 394: 323–334.
- Berends M, Maez R, De Schutter E (2005) The effect of NMDA receptors on gain modulation. *Neural Computation* 17: 2531–2547.
- Bernander O, Douglas RJ, Martin KA, Koch C (1991) Synaptic background activity influences spatiotemporal integration in single pyramidal cells. *Proceedings of the National Academy of Sciences of the USA* 88: 11569–11573.
- Berridge MJ (2006) Calcium microdomains: Organization and function. *Cell Calcium* 40: 405–412.
- Berridge MJ, Bootman MD, Roderick HL (2003) Calcium signalling: Dynamics, homeostasis and remodelling. *Nature Reviews Molecular Cell Biology* 4: 517–529.
- Bezanilla F (1985) Gating of sodium and potassium channels. *Journal of Membrane Biology* 88: 97–111.
- Bhalla US, Bower JM (1993) Exploring parameter space in detailed single neuron models—Simulations of the mitral and granule cells of the olfactory bulb. *Journal of Neurophysiology* 69: 1948–1965.
- Bhalla US, Iyengar R (1999) Emergent properties of networks of biological signaling pathways. *Science* 283: 381–387.
- Bienenstock EL, Cooper LN, Munro PW (1982) Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* 2: 32–48.
- Binzegger T, Douglas RJ, Martin KA (2005) Axons in cat visual cortex are topologically self-similar. *Cerebral Cortex* 15: 152–165.
- Bjaalie JG, Leergaard TB, Lillehaug S, Odeh F, Moene IA, Kjode JO, Darin D (2005) Database and tools for analysis of topographic organization and map transformations in major projection systems of the brain. *Neuroscience* 136: 681–695.
- Blaustein MP, Lederer WJ (1999) Sodium/calcium exchange: Its physiological implications. *Physiological Reviews* 79: 763–854.

- Blinov ML, Faeder JR, Goldstein B, Hlavacek WS (2004) BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics* 20: 3289–3291.
- Bliss TV, Collingridge GL (1993) A synaptic model of memory: Long-term potentiation in the hippocampus. *Nature* 361: 31–39.
- Bliss T, Lomo T (1973) Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology* 232: 331–356.
- Blumenfeld B, Bibitchkov D, Tsodyks M (2006) Neural network model of the primary visual cortex: From functional architecture to lateral connectivity and back. *Journal of Computational Neuroscience* 20: 219–241.
- Blumenfeld H, Zablow L, Sabatini B (1992) Evaluation of cellular mechanisms for modulation of calcium transients using a mathematical model of fura-2 Ca²⁺ imaging in *Aplysia* sensory neurons. *Biophysical Journal* 63: 1146–1164.
- Börgers C, Kopell N (2003) Synchronization in networks of excitatory and inhibitory neurons with sparse, random connectivity. *Neural Computation* 15: 509–538.
- Börgers C, Epstein S, Kopell NJ (2005) Background gamma rhythmicity and attention in cortical local circuits: A computational study. *Proceedings of the National Academy of Sciences of the USA* 102: 7002–7007.
- Borg-Graham L (1991) Modeling the nonlinear conductances of excitable membranes. In H Wheal, J Chad, eds., *Cellular and Molecular Neurobiology: A Practical Approach*, pp. 247–275. New York: Oxford University Press.
- Borg-Graham L (1999) Interpretations of data and mechanisms for hippocampal pyramidal cell models. In E Jones, P Ulinski, A Peters, eds., *Cerebral Cortex*, Vol. 13 *Cortical Models*, pp. 19–138. New York: Plenum.
- Bormann G, Brosens F, De Schutter E (2001) Diffusion. In JM Bower, H Bolouri, eds., *Computational Modeling of Genetic and Biochemical Networks*, pp. 189–222. Cambridge, MA: MIT Press.
- Borst JG, Sakmann B (1998) Calcium current during a single action potential in a large presynaptic terminal of the rat brainstem. *Journal of Physiology* 506: 143–157.
- Bower JM, Beeman D (1998) *The Book of GENESIS: Exploring realistic Neural Models with the GEneral NEural SImulation System*. New York: Telos, Springer-Verlag.
- Bower JM, Bolouri H (2004) *Computational Modeling of Genetic and Biochemical Networks*. Cambridge, MA: MIT Press.
- Brand A (1995) GFP in *Drosophila*. *Trends in Genetics* 11: 324–325.
- Brand AH, Dormand EL (1995) The GAL4 system as a tool for unraveling the mysteries of the *Drosophila* nervous system. *Current Opinion in Neurobiology* 5: 572–578.
- Brémaud A, West DC, Thomson AM (2007) Binomial parameters differ across neocortical layers and with different classes of connections in adult rat and cat neocortex. *Proceedings of the National Academy of Sciences of the USA* 104: 14134–14139.
- Brent R (1976) A new algorithm for minimizing a function of several variables without calculating derivatives. In *Algorithms for Minimization without Derivatives*, pp. 200–248. Englewood Cliffs, NJ: Prentice Hall.
- Brette R (2006) Exact simulation of integrate-and-fire models with synaptic conductances. *Neural Computation* 18: 2004–2027.
- Brette R, Gerstner W (2005) Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology* 94: 3637–3642.
- Brette R, Rudolph M, Carnevale T, Hines M, Beeman D, Bower JM, Diesmann M, Morrison A, Goodman PH, Harris FC Jr, Zirpe M, Natschläger T, Pecevski D, Ermentrout B, Djurfeldt M, Lansner A, Rochel O, Vieville T, Muller E, Davison AP, El Boustani S, Destexhe A (2007) Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience* 23: 349–398.
- Brezina V, Horn CC, Weiss KR (2005) Modeling neuromuscular modulation in *Aplysia*. III. Interaction of central motor commands and peripheral modulatory state for optimal behavior. *Journal of Neurophysiology* 93: 1523–1556.

- Briggman KL, Denk W (2006) Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology* 16: 562–570.
- Brochet DX, Yang D, Di Maio A, Lederer WJ, Franzini-Armstrong C, Cheng H (2005) Ca^{2+} blinks: Rapid nanoscopic store calcium signaling. *Proceedings of the National Academy of Sciences of the USA* 102: 3099–3104.
- Brown KM, Donohue DE, D'Alessandro G, Ascoli GA (2005) A cross-platform freeware tool for digital reconstruction of neuronal arborizations from image stacks. *Neuroinformatics* 3: 343–360.
- Brown TH, Perkel DH, Norris JC, Peacock JH (1981a) Electrotonic structure and specific membrane properties of mouse dorsal root ganglion neurons. *Journal of Neurophysiology* 45: 1–15.
- Brown TH, Fricke RA, Perkel DH (1981b) Passive electrical constants in three classes of hippocampal neurons. *Journal of Neurophysiology* 46: 812–827.
- Broyden CG (1970) The convergence of a class of double-rank minimization algorithms, parts I and II. *IMA Journal of Applied Mathematics* 6: 222–236.
- Brunel N (2000) Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience* 8: 183–208.
- Brunel N, Hakim V (1999) Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Computation* 11: 1621–1671.
- Brunel N, Latham P (2003) Firing rate of noisy quadratic integrate-and-fire neurons. *Neural Computation* 15: 2281–2306.
- Brunel N, van Rossum MC (2007) Lapicque's 1907 paper: From frogs to integrate-and-fire. *Biological Cybernetics* 97: 337–339.
- Brunel N, Wang XJ (2003) What determines the frequency of fast network oscillations with irregular neural discharges? I. Synaptic dynamics and excitation-inhibition balance. *Journal of Neurophysiology* 90: 415–430.
- Brunel N, Chance F, Fourcaud N, Abbott LF (2001) Effects of synaptic noise and filtering on the frequency response of spiking neurons. *Physical Review Letters* 86: 2186–2189.
- Brunel N, Hakim V, Richardson M (2003) Firing rate resonance in a generalized integrate-and-fire neuron with subthreshold resonance. *Physical Review E* 67: 051916.
- Bruno WJ, Yang J, Pearson JE (2005) Using independent open-to-closed transitions to simplify aggregated Markov models of ion channel gating kinetics. *Proceedings of the National Academy of Sciences of the USA* 102: 6326–6331.
- Bucher D, Prinz AA, Marder E (2005) Animal-to-animal variability in motor pattern production in adults and during growth. *Journal of Neuroscience* 25: 1611–1619.
- Buchholtz F, Golowasch J, Epstein IR, Marder E (1992) Mathematical model of an identified stomatogastric ganglion neuron. *Journal of Neurophysiology* 67: 332–340.
- Burdakov D, Petersen OH, Verkhratsky A (2005) Intraluminal calcium as a primary regulator of endoplasmic reticulum function. *Cell Calcium* 38: 303–310.
- Burke RE, Marks WB (2002) Some approaches to quantitative dendritic morphology. In GA Ascoli, ed., *Computational Neuroanatomy: Principles and Methods*, pp. 27–48. Totowa, NJ: Humana.
- Burke RE, ten Bruggencate G (1971) Electrotonic characteristics of motoneurons of varying size. *Journal of Physiology* 212: 1–20.
- Burke RE, Fyffe REW, Moschovakis AK (1994) Electrotonic architecture of cat gamma motoneurons. *Journal of Neurophysiology* 72: 2302–2316.
- Burke RE, Marks WB, Ulfhake B (1992) A parsimonious description of motoneurons' dendritic morphology using computer simulation. *Journal of Neuroscience* 12: 2403–2416.
- Burkitt AN (2006a) A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. *Biological Cybernetics* 95: 1–19.
- Burkitt AN (2006b) A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties. *Biological Cybernetics* 95: 97–112.

- Burkitt AN, Meffin H, Grayden DB (2004) Spike-timing-dependent plasticity: The relationship to rate-based learning for models with weight dynamics determined by a stable fixed point. *Neural Computation* 16: 885–940.
- Burnashev N (1998) Calcium permeability of ligand-gated channels. *Cell Calcium* 24: 325–332.
- Burrage K, Tian TH (2004) Poisson Runge-Kutta methods for chemical reaction systems. In Y Lu, W Sun, T Tang, eds., *Advances in Scientific Computing and Applications*, pp. 82–96. Beijing: Science Press.
- Burrage K, Tian TH, Burrage P (2004) A multi-scaled approach for simulating chemical reaction systems. *Progress in Biophysics and Molecular Biology* 85: 217–234.
- Bush P, Sejnowski TJ (1991) Simulations of a reconstructed cerebellar Purkinje cell based on simplified channel kinetics. *Neural Computation* 3: 321–332.
- Buzsáki G (2006) *Rhythms of the Brain*. New York: Oxford University Press.
- Cais O, Sedlacek M, Horak M, Dittert I, Vyklický L (2008) Temperature dependence of NR1/NR2b NMDA receptor channels. *Neuroscience* 151: 428–438.
- Calabrese RL (1995) Oscillation in motor pattern-generating networks. *Current Opinion in Neurobiology* 5: 816–823.
- Calcraft L, Adams R, Davey N (2008) Efficient connection strategies in 1D and 2D associative memory models with and without displaced connectivity. *BioSystems* 94: 87–94.
- Cang JJ, Friesen WO (2002) Model for intersegmental coordination of leech swimming: Central and sensory mechanisms. *Journal of Neurophysiology* 87: 2760–2769.
- Cannon RC and D'Alessandro G (2006) The ion channel inverse problem: Neuroinformatics meets biophysics. *PLoS Computational Biology* 2: e91.
- Cannon RC, Turner DA, Pyapali GK, Wheal HV (1998) An on-line archive of reconstructed hippocampal neurons. *Journal of Neuroscience Methods* 84: 49–54.
- Cannon RC, Howell FW, Goddard NH, De Schutter E (2002) Non-curated distributed databases for experimental data and models in neuroscience. *Network: Computation in Neural Systems* 13: 415–428.
- Cannon RC, Gewaltig M-O, Gleeson P, Bhalla US, Cornelis H, Hines ME, Howell FW, Muller E, Stiles JR, Wils S, De Schutter E (2007) Interoperability of neuroscience modeling software: Current status and future directions. *Neuroinformatics* 5: 127–138.
- Cannon SC, Robinson DA (1985) An improved neural-network model for the neural integrator of the oculomotor system: More realistic neuron behavior. *Biological Cybernetics* 53: 93–108.
- Cao Y, Li H, Petzold L (2004) Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *Journal of Chemical Physics* 121: 4059–4067.
- Cao Y, Gillespie DT, Petzold LR (2005a) Avoiding negative populations in explicit Poisson tau-leaping. *Journal of Chemical Physics* 123: 054104.
- Cao Y, Gillespie DT, Petzold LR (2005b) The slow-scale stochastic simulation algorithm. *Journal of Chemical Physics* 122: 014116.
- Cao Y, Gillespie DT, Petzold LR (2006) Efficient step size selection for the tau-leaping simulation method. *Journal of Chemical Physics* 124: 044109.
- Capaday C, van Vreeswijk C (2006) Direct control of firing rate gain by dendritic shunting inhibition. *Journal of Integrative Neuroscience* 5: 199–222.
- Caporale N, Dan Y (2008) Spike timing-dependent plasticity: A Hebbian learning rule. *Annual Review of Neuroscience* 31: 25–46.
- Capowski JJ (1985) *Computer Techniques in Neuroanatomy*. New York: Plenum.
- Carnevale NT, Hines ML (2006) *The NEURON Book*. Cambridge, UK: Cambridge University Press.
- Carnevale NT, Rosenthal S (1992) Kinetics of diffusion in a spherical cell. I. No solute buffering. *Journal of Neuroscience Methods* 41: 205–216.
- Carnevale NT, Tsai KY, Claiborne BJ, Brown TH (1997) Comparative electrotonic analysis of three classes of rat hippocampus neurons. *Journal of Neurophysiology* 78: 703–720.
- Carpenter DO, Hovey MM, Bak AF (1975) Resistivity of axoplasm. II. Internal resistivity of giant axons of squid and myxicola. *Journal of General Physiology* 66: 139–148.

- Cathala L, Misra C, Cull-Candy S (2000) Developmental profile of the changing properties of NMDA receptors at cerebellar mossy fiber-granule cell synapses. *Journal of Neuroscience* 20: 5899–5905.
- Cathala L, Holderith NB, Nusser Z, DiGregorio DA, Cull-Candy SG (2005) Changes in synaptic structure underlie the developmental speeding of AMPA receptor-mediated EPSCs. *Nature Neuroscience* 8: 1310–1318.
- Chalfie M, Tu Y, Euskirchen G, Ward WW, Prasher DC (1994) Green fluorescent protein as a marker for gene expression. *Science* 263: 802–805.
- Chance FS, Nelson SB, Abbott LF (1998) Synaptic depression and the temporal response characteristics of V1 cells. *Journal of Neuroscience* 18: 4785–4799.
- Chance FS, Abbott LF, Reyes AD (2002) Gain modulation from background synaptic input. *Neuron* 35: 773–782.
- Chatterjee A, Vlachos DG, Katsoulakis MA (2005) Binomial distribution-based tau-leap accelerated stochastic simulation. *Journal of Chemical Physics* 122: 024112.
- Chávez M, Le Van Quyen M, Navarro V, Baulac M, Martinic J (2003) Spatio-temporal dynamics prior to neocortical seizures: Amplitude versus phase couplings. *IEEE Transactions in Biomedical Engineering* 50: 571–583.
- Chen C, Hess P (1990) Mechanisms of gating of T-type calcium channels. *Journal of General Physiology* 96: 603–630.
- Chiel HJ, Beer RD (1997) The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neurosciences* 20: 553–557.
- Chitwood RA, Hubbard A, Jaffe DB (1999) Passive electrotonic properties of rat hippocampal CA3 interneurons. *Journal of Physiology* 515: 743–756.
- Choi YM, Kim SH, Chung S, Uhm DY, Park MK (2006) Regional interaction of endoplasmic reticulum Ca²⁺ signals between soma and dendrites through rapid luminal Ca²⁺ diffusion. *Journal of Neuroscience* 26: 12127–12136.
- Claiborne BJ (1992) Use of computers for quantitative, three-dimensional analysis of dendritic trees. In PM Conn, ed., *Methods in Neurosciences*, Vol. 10 *Computers and Computations in the Neurosciences*, pp. 315–330. San Diego: Academic Press.
- Claiborne BJ, Amaral DG, Cowan WM (1986) A light and electron microscopic analysis of the mossy fibers of the dentate gyrus. *Journal of Comparative Neurology* 246: 435–458.
- Claiborne BJ, Amaral DG, Cowan WM (1990) Quantitative three-dimensional analysis of granule cell dendrites in the rat dentate gyrus. *Journal of Comparative Neurology* 302: 206–219.
- Clark BA, Cull-Candy SG (2002) Activity-dependent recruitment of extrasynaptic NMDA receptor activation at an AMPA receptor-only synapse. *Journal of Neuroscience* 22: 4428–4436.
- Clark BA, Monsivais P, Branco T, London M, Häusser M (2005) The site of action potential initiation in cerebellar Purkinje neurons. *Nature Neuroscience* 8: 137–139.
- Clements JD, Redman SJ (1989) Cable properties of cat spinal motoneurons measured by combining voltage clamp, current clamp and intracellular staining. *Journal of Physiology* 409: 63–87.
- Coello Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191: 1245–1287.
- Cole KS (1968) *Membranes, Ions and Impulses*. Berkeley, CA: University of California Press.
- Cole KS (1975) Resistivity of axoplasm. I. Resistivity of extruded squid axoplasm. *Journal of General Physiology* 66: 133–138.
- Colquhoun D, Hawkes AG (1981) On the stochastic properties of single ion channels. *Proceedings of the Royal Society London Series B* 211: 205–235.
- Compte A (2006) Computational and in vitro studies of persistent activity: Edging toward cellular and synaptic mechanisms of working memory. *Neuroscience* 139: 135–151.
- Cook EP, Johnston D (1999) Voltage-dependent properties of dendrites that eliminate location-dependent variability of synaptic input. *Journal of Neurophysiology* 81: 535–543.

- Cornelis H, De Schutter E (2003) Neurospaces: Separating modeling and simulation. *Neurocomputing* 52–54: 227–231.
- Cornelis H, De Schutter E (2007) Neurospaces: Toward automated model partitioning for parallel computers. *Neurocomputing* 70: 2117–2121.
- Cornelisse LN, van Elburg RA, Meredith RM, Yuste R, Mansvelder HD (2007) High speed two-photon imaging of calcium dynamics in dendritic spines: consequences for spine calcium kinetics and buffer capacity. *PLoS ONE* 2: e1073.
- Cornish-Bowden A (2004) *Fundamentals of Enzyme Kinetics*, 3rd ed. London: Portland Press.
- Cortassa S, Aon MA, Marban E, Winslow RL, O'Rourke B (2003) An integrated model of cardiac mitochondrial energy metabolism and calcium dynamics. *Biophysical Journal* 84: 2734–2755.
- Coulter DA, Huguenard JR, Prince DA (1989) Calcium currents in rat thalamocortical relay neurones: Kinetic properties of the transient, low-threshold current. *Journal of Physiology* 414: 587–604.
- Coutinho V, Knöpfel T (2002) Metabotropic glutamate receptors: Electrical and chemical signaling properties. *Neuroscientist* 8: 551–561.
- Cowan JD (1974) Mathematical models of large-scale nervous activity. In *Lectures on Mathematics in the Life Sciences*, Vol. 6, pp. 99–133, Providence, RI: American Mathematical Society.
- Crook SM, Ermentrout GB, Bower JM (1998) Spike frequency adaptation affects the synchronization properties of networks of cortical oscillations. *Neural Computation* 10: 837–854.
- Cull-Candy SG, Leszkiewicz DN (2004) Role of distinct NMDA receptor subtypes at central synapses. *Science STKE* 255: re16.
- Cummins GI, Crook SM, Dimitrov AG, Ganje T, Jacobs GA, Miller JP (2003) Structural and biophysical mechanisms underlying dynamic sensitivity of primary sensory interneurons in the cricket cercal sensory system. *Neurocomputing* 52–54: 45–52.
- Curtis HJ, Cole KS (1938) Transverse electric impedance of the squid giant axon. *Journal of General Physiology* 21: 757–765.
- Cymbalyuk GS, Gaudry Q, Masino MA, Calabrese RL (2002) Bursting in leech heart interneurons: Cell-autonomous and network-based mechanisms. *Journal of Neuroscience* 22: 10580–10592.
- Dan Y, Poo MM (2006) Spike timing-dependent plasticity: From synapse to perception. *Physiological Reviews* 86: 1033–1048.
- Davison AP, Brüderle D, Eppler J, Kremkow J, Müller E, Pecevski D, Perrinet L, Yger P (2009) PyNN: A common interface for neuronal network simulators. *Frontiers in Neuroinformatics* 2: 11.
- Davison AP, Feng J, Brown D (2000) A reduced compartmental model of the mitral cell for use in network models of the olfactory bulb. *Brain Research Bulletin* 51: 393–399.
- Dayan P, Abbott LF (2002) *Theoretical Neuroscience*. Cambridge, MA: MIT Press.
- De Schutter E (1998) Dendritic voltage and calcium-gated channels amplify the variability of postsynaptic responses in a Purkinje cell model. *Journal of Neurophysiology* 80: 504–519.
- De Schutter E (2008) Why are computational neuroscience and systems biology so separate? *PLoS Computational Biology* 4: e1000078.
- De Schutter E, Beeman D (1998) Speeding up GENESIS simulations. In JM Bower, D Beeman, eds., *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*, pp. 329–347. New York: Telos, Springer-Verlag.
- De Schutter E, Bower JM (1994a) An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice. *Journal of Neurophysiology* 71: 375–400.
- De Schutter E, Bower JM (1994b) An active membrane model of the cerebellar Purkinje cell: II. Simulation of synaptic responses. *Journal of Neurophysiology* 71: 401–419.
- De Schutter E, Bower JM (1994c) Simulated responses of cerebellar Purkinje cells are independent of the dendritic location of granule cell synaptic inputs. *Proceedings of the National Academy of Sciences of the USA* 91: 4736–4740.
- De Schutter E, Smolen P (1998) Calcium dynamics in large neuronal models. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 211–250. Cambridge, MA: MIT Press.

- De Schutter E, Angstadt JD, Calabrese RL (1993) A model of graded synaptic transmission for use in dynamic network simulations. *Journal of Neurophysiology* 69: 1225–1235.
- De Schutter E, Ekeberg O, Kotaleski JH, Achard P, Lansner A (2005) Biophysically detailed modelling of microcircuits and beyond. *Trends in Neurosciences* 28: 562–569.
- De Young GW, Keizer J (1992) A single-pool inositol 1,4,5-triphosphate-receptor-based model for agonist-stimulated oscillations in Ca^{2+} concentration. *Proceedings of the National Academy of Sciences of the USA* 89: 9895–9899.
- Deb K (2001) *Multi-objective Optimization using Evolutionary Algorithms*. Chichester, UK: Wiley.
- Debanne D (2004) Information processing in the axon. *Nature* 5, 304–316.
- Del Castillo J, Katz B (1954) Quantal components of the end-plate potential. *Journal of Physiology* 124: 560–573.
- Desmond NL, Levy WB (1982) A quantitative anatomical study of the granule cell dendritic fields of the rat dentate gyrus using a novel probabilistic method. *Journal of Comparative Neurology* 212: 131–145.
- Destexhe A, Huguenard JR (2000) Nonlinear thermodynamic models of voltage-dependent currents. *Journal of Computational Neuroscience* 9: 259–270.
- Destexhe A, Pare D (1999) Impact of network activity on the integrative properties of neocortical pyramidal neurons in vivo. *Journal of Neurophysiology* 81: 1531–1547.
- Destexhe A, Rudolph M (2004) Extracting information from the power spectrum of synaptic noise. *Journal of Computational Neuroscience* 17: 327–345.
- Destexhe A, Sejnowski TJ (2003) Interactions between membrane conductances underlying thalamocortical slow-wave oscillations. *Physiological Reviews* 83: 1401–1453.
- Destexhe A, Mainen ZF, Sejnowski TJ (1994) Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism. *Journal of Computational Neuroscience* 1: 195–230.
- Destexhe A, Contreras D, Steriade M, Sejnowski TJ, Huguenard JR (1996) In vivo, in vitro and computational analysis of dendritic calcium currents in thalamic reticular neurons. *Journal of Neuroscience* 16: 169–185.
- Destexhe A, Mainen ZF, Sejnowski TJ (1998a) Kinetic models of synaptic transmission. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 1–26. Cambridge, MA: MIT Press.
- Destexhe A, Neubig M, Ulrich D, Huguenard JR (1998b) Dendritic low-threshold calcium currents in thalamic relay cells. *Journal of Neuroscience* 18: 3574–3588.
- Destexhe A, Rudolph M, Fellous JM, Sejnowski TJ (2001) Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. *Neuroscience* 107: 13–24.
- Devaud JM, Quenet B, Gascuel J, Masson C (2000) Statistical analysis and parsimonious modelling of dendograms of in vitro neurons. *Bulletin of Mathematical Biology* 62: 657–674.
- Devroye L (1986) *Non-uniform Random Variate Generation*. New York: Springer-Verlag.
- Diamond JS (2001) Neuronal glutamate transporters limit activation of NMDA receptors by neurotransmitter spillover on CA1 pyramidal cells. *Journal of Neuroscience* 21: 8328–8338.
- Diamond JS (2005) Deriving the glutamate clearance time course from transporter currents in CA1 hippocampal astrocytes: Transmitter uptake gets faster during development. *Journal of Neuroscience* 25: 2906–2916.
- Diba K, Koch C, Segev I (2006) Spike propagation in dendrites with stochastic ion channels. *Journal of Computational Neuroscience* 20: 77–84.
- Diesmann M, Gewaltig MO, Aertsen A (1999) Stable propagation of synchronous spiking in cortical neural networks. *Nature* 402: 529–533.
- DiFrancesco D, Noble D (1985) A model of cardiac electrical activity incorporating ionic pumps and concentration changes. *Philosophical Transactions of the Royal Society of London, Series B* 307: 353–398.
- DiGregorio DA, Rothman JS, Nielsen TA, Silver RA (2007) Desensitization properties of AMPA receptors at the cerebellar mossy fiber granule cell synapse. *Journal of Neuroscience* 27: 8344–8357.

- Dipolo R, Beaugé L (2006) Sodium/calcium exchanger: Influence of metabolic regulation on ion carrier interactions. *Physiological Reviews* 86: 155–203.
- Djurfeldt M, Johansson C, Ekeberg Ö, Rehn M, Lundqvist M, Lansner A (2005) Massively parallel simulation of brain-scale neuronal network models. Technical Report TRITA-NA-P0513 KTH, School of Computer Science and Communication, Stockholm. http://www-03.ibm.com/servers/deepcomputing/pdf/Blue_Gene_Applications_Paper_KTH_0306.pdf.
- Dode L, Vilsen B, Van Baelen K, Wuytack F, Clausen JD, Andersen JP (2002) Dissection of the functional differences between sarco(endo)plasmic reticulum Ca^{2+} -ATPase (SERCA) 1 and 3 isoforms by steady-state and transient kinetic analyses. *Journal of Biological Chemistry* 277: 45579–45591.
- Dodla R, Svirskis G, Rinzel J (2006) Well-timed, brief inhibition can promote spiking: Postinhibitory facilitation. *Journal of Neurophysiology* 95: 2664–2677.
- Doi T, Kuroda S, Michikawa T, Kawato M (2005) Inositol 1,4,5-trisphosphate-dependent Ca^{2+} threshold dynamics detect spike timing in cerebellar Purkinje cells. *Journal of Neuroscience* 25: 950–961.
- Donohue DE, Scorcioni R, Ascoli GA (2002) Generation and description of neuronal morphology using L-Neuron: A case study. In GA Ascoli, ed., *Computational Neuroanatomy: Principles and Methods*, pp. 49–70. Totowa, NJ: Humana.
- Dorval AD, White JA (2006) Synaptic input statistics tune the variability and reproducibility of neuronal responses. *Chaos* 16: 026105.
- Dorval AD, Christini DJ, White JA (2001) Real-time linux dynamic clamp: A fast and flexible way to construct virtual ion channels in living cells. *Annals of Biomedical Engineering* 29: 897–907.
- Druckmann S, Banitt Y, Gidon A, Schürmann F, Markram H, Segev I (2007) A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data. *Frontiers in Neuroscience* 1: 7–18.
- Duan H, Wearne SL, Morrison JH, Hof PR (2002) Quantitative analysis of the dendritic morphology of corticocortical projection neurons in the macaque monkey association cortex. *Neuroscience* 114: 349–359.
- Duan H, Wearne SL, Rocher AB, Macedo A, Morrison JH, Hof PR (2003) Age-related dendritic and spine changes in corticocortically projecting neurons in macaque monkeys. *Cerebral Cortex* 13: 950–961.
- Durand D, Carlen PL, Gurevich N, Ho A, Kunov H (1983) Electrotonic parameters of rat dentate granule cells measured using short current pulses and HRP staining. *Journal of Neurophysiology* 50: 1080–1097.
- Durstewitz D, Gabriel T (2007) Dynamical basis of irregular spiking in NMDA-driven prefrontal cortex neurons. *Cerebral Cortex* 17: 894–908.
- Durstewitz D, Seamans JK (2006) Beyond bistability: Biophysics and temporal dynamics of working memory. *Neuroscience* 139: 119–133.
- Dyhrfjeld-Johnsen J, Maier J, Schubert D, Staiger J, Luhmann HJ, Stephan KE, Kötter R (2005) CoCo-Dat: A database system for organizing and selecting quantitative data on single neurons and neuronal microcircuitry. *Journal of Neuroscience Methods* 141: 291–308.
- Edelstein-Keshet L (2005) *Mathematical Models in Biology*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Eiben AE, Smith JE (2003) *Introduction to Evolutionary Computing*. Berlin: Springer-Verlag.
- Ekeberg Ö (2001) Modeling of interactions between neural networks and musculoskeletal systems. In E De Schutter, ed., *Computational Neuroscience: Realistic Modeling for Experimentalists*, pp. 317–335. Boca Raton, FL: CRC Press.
- Ekeberg Ö, Grillner S (1999) Simulations of neuromuscular control in lamprey swimming. *Philosophical Transactions of the Royal Society of London, Series B* 354: 895–902.
- Ekeberg Ö, Blumel M, Buschges A (2004) Dynamic simulation of insect walking. *Arthropod Structure and Development* 33: 287–300.
- Elf J, Doncic A, Ehrenberg M (2003) Mesoscopic reaction-diffusion in intracellular signaling. *Proceedings of SPIE* 5110: 114–124.
- Eliasmith C, Anderson CH (2003) *Neural Engineering: Computation, Representation and Dynamics in Neu-robiological Systems*. Cambridge, MA: MIT Press.

- Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403: 335–338.
- Engelbrecht AP (2005) *Fundamentals of Computational Swarm Intelligence*. Chichester, UK: Wiley.
- Ermentrout GB (1996) Type I membranes, phase resetting curves, and synchrony. *Neural Computation* 8: 979–1001.
- Ermentrout GB (1998) Linearization of F–I curves by adaptation. *Neural Computation* 10: 1721–1729.
- Ermentrout GB, Kleinfeld D (2001) Traveling electrical waves in cortex: Insights from phase dynamics and speculation on a computational role. *Neuron* 29: 33–44.
- Ermentrout GB, Kopell N (1986) Parabolic bursting in an excitable system coupled with a slow oscillation. *SIAM Journal on Applied Mathematics* 46: 233–253.
- Ermentrout GB, Pascal M, Gutkin B (2001) The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators. *Neural Computation* 13: 1285–310.
- Ermis ZS, Antal K, Gulyas AI, Megias M, Freund TF (2001) Electrotonic profile and passive propagation of synaptic potentials in three subpopulations of hippocampal CA1 interneurons. *Neuroscience* 104: 1013–1026.
- Esser SK, Hill SL, Tononi G (2007) Sleep homeostasis and cortical synchronization: I. Modeling the effects of synaptic strength on sleep slow waves. *Sleep* 30: 1617–1630.
- Eyring H (1935) The activated complex in chemical reactions. *Journal of Chemical Physics* 3: 107–115.
- Faas GC, Schwaller B, Vergara JL, Mody I (2007) Resolving the fast kinetics of cooperative binding: Ca^{2+} buffering by calretinin. *PLoS Biology* 5: e311.
- Faisal AA, Laughlin SB (2007) Stochastic simulations on the reliability of action potential propagation in thin axons. *PLoS Computational Biology* 3: e79.
- Faisal AA, Selen LPJ, Wolpert DM (2008). Noise in the nervous system. *Nature Reviews Neuroscience* 9: 292–303.
- Fakler B, Adelman JP (2008) Control of K(Ca) channels by calcium nano/microdomains. *Neuron* 59: 873–881.
- Fall CP, Marland ES, Wagner JM, Tyson JJ (eds.) (2005) *Computational Cell Biology*. New York: Springer.
- Fan RJ, Marin-Burgin A, French KA, Friesen OW (2005) A dye mixture (Neurobiotin and Alexa 488) reveals extensive dye-coupling among neurons in leeches; physiology confirms the connections. *Journal of Comparative Physiology A* 12: 1157–1171
- Feldmeyer D, Cull-Candy SG (1996) Functional consequences of changes in NMDA receptor subunit expression during development. *Journal of Neurocytology* 25: 857–867.
- Feldmeyer D, Lubke J, Silver RA, Sakmann B (2002) Synaptic connections between layer 4 spiny neurone-layer 2/3 pyramidal cell pairs in juvenile rat barrel cortex: Physiology and anatomy of interlaminar signalling within a cortical column. *Journal of Physiology* 538: 803–822.
- Fellous JM, Rudolph M, Destexhe A, Sejnowski TJ (2003) Synaptic background noise controls the input/output characteristics of single cells in an in vitro model of in vivo activity. *Neuroscience* 122: 811–829.
- Feng G, Mellor RH, Bernstein M, Keller-Peck C, Nguyen QT, Wallace M, Nerbonne JM, Lichtman JW, Sanes JR (2000) Imaging neuronal subsets in transgenic mice expressing multiple spectral variants of GFP. *Neuron* 28: 41–51.
- Feng J, Li G (2002) Impact of geometrical structures on the output of neuronal models: A theoretical and numerical analysis. *Neural Computation* 14: 621–640.
- Fernandez FR, Engbers JD, Turner RW (2007) Firing dynamics of cerebellar Purkinje cells. *Journal of Neurophysiology* 98: 278–294.
- Ferziger JH, Peric M (2001) *Computational Methods for Fluid Dynamics*. Berlin-Heidelberg: Springer-Verlag.
- Fiacco TA, McCarthy KD (2006) Astrocyte calcium elevations: Properties, propagation, and effects on brain signaling. *Glia* 54: 676–690.

- Fiala, JC (2005) Reconstruct: A free editor for serial section microscopy. *Journal of Microscopy* 218: 52–61.
- Fiala JC, Harris KM (2001) Cylindrical diameters method for calibrating section thickness in serial electron microscopy. *Journal of Microscopy* 202: 468–472.
- Fiala JC, Harris KM (2002) Computer-based alignment and reconstruction of serial sections. *Microscopy and Analysis* 52: 5–7.
- Fill M, Copello JA (2002) Ryanodine receptor calcium release channels. *Physiological Reviews* 82: 893–922.
- Finkel DE (2003) *DIRECT Optimization Algorithm User Guide*. Technical report crsc-tr03-11, North Carolina State University, Raleigh.
- FitzHugh R (1961) Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal* 1: 445–466.
- Fitzhugh R (1965) A kinetic model of the conductance changes in nerve membrane. *Journal of Cellular and Comparative Physiology* 66: 111–118.
- Fiumelli H, Woodin MA (2007) Role of activity-dependent regulation of neuronal chloride homeostasis in development. *Current Opinion in Neurobiology* 17: 81–86.
- Fleshman JW, Segev I, Burke RE (1988) Electrotonic architecture of type-identified alpha-motoneurons in the cat spinal cord. *Journal of Neurophysiology* 60: 60–85.
- Fletcher CAJ (1991) *Computational Techniques for Fluid Dynamics*, Vol. I. Berlin: Springer-Verlag.
- Fletcher R (1970) A new approach to variable metric algorithms. *Computer Journal* 13: 317–322.
- Fletcher R, Powell MJD (1963) A rapidly convergent descent method for minimization. *Computer Journal* 6: 163–168.
- Foldy C, Aradi I, Howard A, Soltesz I (2004) Diversity beyond variance: Modulation of firing rates and network coherence by GABAergic subpopulations. *European Journal of Neuroscience* 19: 119–130.
- Foster WR, Ungar LH, Schwaber JS (1993) Significance of conductances in Hodgkin-Huxley models. *Journal of Neurophysiology* 70: 2502–2518.
- Fourcaud-Trocmé N, Hansel D, van Vreeswijk C, Brunel N (2003) How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of Neuroscience* 23: 11628–11640.
- Frandsen PE, Jonasson K, Nielsen HB, Tingleff O (2004) Unconstrained Optimization. Lyngby, Denmark: Department of Mathematical Modeling, Technical University of Denmark. <http://www.imm.dtu.dk/courses/02611/uncon.pdf>
- Franks KM, Stevens CF, Sejnowski TJ (2003) Independent sources of quantal variability at single glutamatergic synapses. *Journal of Neuroscience* 23: 3186–3195.
- Freeman WJ (1975) *Mass Action in the Nervous System*. New York: Academic Press.
- Friel DD (2000) Mitochondria as regulators of stimulus-evoked calcium signals in neurons. *Cell Calcium* 28: 307–316.
- Friesen WO, Cang JH (2001) Sensory and central mechanisms control intersegmental coordination. *Current Opinion in Neurobiology* 11: 678–683.
- Friesen WO, Block GD, Hocker CG (1993) Formal approaches to understanding biological oscillators. *Annual Review of Physiology* 55: 661–681.
- Fuhrmann G, Markram H, Tsodyks M (2002) Spike frequency adaptation and neocortical rhythms. *Journal of Neurophysiology* 88: 761–770.
- Fusi S, Mattia M (1999) Collective behavior of networks with linear (VLSI) integrate and fire neurons. *Neural Computation* 11: 633–652.
- Gabbiani F, Koch C (1998) Principles of spike train analysis. In C Koch, I Segev, eds., *Methods in Neural Modeling: From Ions to Networks*, pp. 313–360. Cambridge, MA: MIT Press.
- Gardner D, Shepherd GM (2004) A gateway to the future of neuroinformatics. *Neuroinformatics* 2: 271–274.
- Garza-Meilandt A, Cantu RE, Claiborne BJ (2006) Estradiol's effects on learning and neuronal morphology vary with route of administration. *Behavioral Neuroscience* 120: 905–916.

- Gasparini S, Migliore M, Magee JC (2004) On the initiation and propagation of dendritic spikes in CA1 pyramidal neurons. *Journal of Neuroscience* 24: 11046–11056.
- Geiger JR, Melcher T, Koh DS, Sakmann B, Seburg PH, Jonas P, Monyer H (1995) Relative abundance of subunit mRNAs determines gating and Ca^{2+} permeability of AMPA receptors in principal neurons and interneurons in rat CNS. *Neuron* 15: 193–204.
- Geiger JR, Lübbe J, Roth A, Frotscher M, Jonas P (1997) Submillisecond AMPA receptor-mediated signaling at a principal neuron-interneuron synapse. *Neuron* 18: 1009–1023.
- Genoud C, Quairiaux C, Steiner P, Hirning H, Welker E, Knott, GW (2006) Plasticity of astrocytic coverage and glutamate transporter expression in adult mouse cortex. *PLoS Biology* 4: e343.
- Gentet LJ, Stuart GJ, Clements JD (2000) Direct measurement of specific membrane capacitance in neurons. *Biophysical Journal* 79: 314–320.
- Gerstner W. (1995) Time structure of the activity in neural network models. *Physical Review E* 51: 738–758.
- Gerstner W, Kistler WM (2002) *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, UK: Cambridge University Press.
- Gerstner W, Ritz R, van Hemmen JL (1993) Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological Cybernetics* 69: 503–515.
- Gerstner W, van Hemmen JL, Cowan JD (1996) What matters in neuronal locking? *Neural Computation* 8: 1653–1676.
- Getting PA (1989) Reconstruction of small neural networks. In C Koch, I Segev, eds., *Methods in Neural Modeling: From Synapses to Networks*, pp. 171–190. Cambridge, MA: MIT Press.
- Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A* 104: 1876–1889.
- Gillespie DT (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical species. *Journal of Computational Physics* 22: 403–434.
- Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry A* 81: 2340–2361.
- Gillespie DT (2000) The chemical Langevin equation. *Journal of Chemical Physics* 113: 297–306.
- Gillespie DT (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics* 115: 1716–1733.
- Giugliano M (2000) Synthesis of generalized algorithms for the fast computation of synaptic conductances with Markov kinetic models in large network simulations. *Neural Computation* 12: 903–931.
- Giugliano M, Bove M, Grattarola M (1999) Fast calculation of short-term depressing synaptic conductances. *Neural Computation* 11: 1413–1426.
- Glaser JR, Glaser EM (1990) Neuron imaging with NeuroLucida—A PC-based system for image combining microscopy. *Computerized Medical Imaging and Graphics* 14: 307–317.
- Gleeson P, Steuber V, Silver RA (2007) neuroConstruct: A tool for modeling networks of neurons in 3D space. *Neuron* 54: 219–235.
- Goaillard JM, Marder E (2006) Dynamic clamp analyses of cardiac, endocrine, and neural function. *Physiology* 21: 197–207.
- Goddard NH, Hucka M, Howell F, Cornelis H, Shankar K, Beeman D (2001) Towards NeuroML: Model description methods for collaborative modelling in neuroscience. *Philosophical Transactions of the Royal Society of London, Series B* 356: 1209–1228.
- Gold C, Henze DA, Koch C, Buzsáki G (2006) On the origin of the extracellular action potential waveform: A modeling study. *Journal of Neurophysiology* 95: 3113–3128.
- Gold C, Henze DA, Koch C (2007) Using extracellular action potential recordings to constrain compartmental models. *Journal of Computational Neuroscience* 23: 39–58.
- Goldberg JA, Wilson CJ (2005) Control of spontaneous firing patterns by the selective coupling of calcium currents to calcium-activated potassium currents in striatal cholinergic interneurons. *Journal of Neuroscience* 25: 10230–10238.

- Goldbeter A, Dupont G, Berridge MJ (1990) Minimal model for signal-induced Ca^{2+} oscillations and for their frequency encoding through protein phosphorylation. *Proceedings of the National Academy of Sciences of the USA* 87: 1461–1465.
- Goldfarb D (1970) A family of variable metric methods derived by variational means. *Mathematics of Computation* 24: 23–26.
- Golding NL, Mickus TJ, Katz Y, Kath WL, Spruston N (2005) Factors mediating powerful voltage attenuation along CA1 pyramidal neuron dendrites. *Journal of Physiology* 568: 69–82.
- Goldman MS, Golowasch J, Marder E, Abbott LF (2001) Global structure, robustness, and modulation of neuronal models. *Journal of Neuroscience* 21: 5229–5238.
- Goldman MS, Maldonado P, Abbott LF (2002) Redundancy reduction and sustained firing with stochastic depressing synapses. *Journal of Neuroscience* 22: 584–591.
- Golomb D, Rinzel J (1993) Dynamics of globally coupled inhibitory neurons with heterogeneity. *Physical Review E* 48: 4810–4814.
- Golowasch J, Buchholtz F, Epstein IR, Marder E (1992) Contribution of individual ionic currents to activity of a model stomatogastric ganglion neuron. *Journal of Neurophysiology* 67: 341–349.
- Golowasch J, Abbott LF, Marder E (1999) Activity-dependent regulation of potassium currents in an identified neuron of the stomatogastric ganglion of the crab *Cancer borealis*. *Journal of Neuroscience* 19: RC33.
- Golowasch J, Goldman MS, Abbott LF, Marder E (2002) Failure of averaging in the construction of a conductance-based neuron model. *Journal of Neurophysiology* 87: 1129–1131.
- Goutsias J (2005) Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *Journal of Chemical Physics* 122: 184102.
- Graupner M, Brunel N (2007) STDP in a bistable synapse model based on CaMKII and associated signaling pathways. *PLoS Computational Biology* 3: e221.
- Griffith JS (1963) On the stability of brain-like structures. *Biophysical Journal* 3: 299–308.
- Grillner S, Markram H, De Schutter E, Silberberg G, LeBeau FE (2005) Microcircuits in action—from CPGs to neocortex. *Trends in Neurosciences* 28: 525–533.
- Groc L, Gustafsson B, Hanse E (2006) AMPA signalling in nascent glutamatergic synapses: There and not there! *Trends in Neurosciences* 29: 132–139.
- Grynkiewicz G, Poenie M, Tsien RYA (1985) A new generation of Ca^{2+} indicators with greatly improved fluorescence properties. *Journal of Biological Chemistry* 260: 3440–3450.
- Guerrero-Rivera R, Morrison A, Diesmann M, Pearce TC (2006) Programmable logic construction kits for hyper-real-time neuronal modeling. *Neural Computation* 18: 2651–2679.
- Günay C, Edgerton JR, Jaeger D (2008) Channel density distributions explain spiking variability in the globus pallidus: A combined physiology and computer simulation database approach. *Journal of Neuroscience* 28: 7476–7491.
- Gutkin B, Ermentrout GB (1998) Dynamics of membrane excitability determine interspike interval variability: A link between spike generation mechanisms and cortical spike train statistics. *Neural Computation* 10: 1047–1065.
- Hansel D, Mato G, Meunier C, Neltner L (1998) On numerical simulations of integrate-and-fire neural networks. *Neural Computation* 10: 467–483.
- Harris KM, Perry E, Bourne J, Feinberg M, Ostroff L, Hurlburt J (2006) Uniform serial sectioning for transmission electron microscopy. *Journal of Neuroscience* 26: 12101–12103.
- Harris KM, Stevens JK (1988) Dendritic spine of rat cerebellar Purkinje cells: Serial electron microscopy with reference to their biophysical characteristics. *Journal of Neuroscience* 8: 4455–4469.
- Harsch A, Robinson HP (2000) Postsynaptic variability of firing in rat cortical neurons: The roles of input synchronization and synaptic NMDA receptor conductance. *Journal of Neuroscience* 20: 6181–6192.
- HarshaRani GV, Vayttaden SJ, Bhalla US (2005) Electronic data sources for kinetic models of cell signaling. *Journal of Biochemistry* 137: 653–657.
- Hartline KK (1967) <http://nobelprize.org/medicine/laureates/1967/hartline-lecture.html>.

- Hartline KK, Ratliff F (1972) Inhibitory interactions in the retina of *Limulus*. In MGF Fuortes, ed., *Handbook of Sensory Physiology*, Vol. VII/2 *Physiology of Photoreceptor Organs*, pp. 381–447. Berlin: Springer-Verlag.
- Hartmann J, Konnerth A (2005) Determinants of postsynaptic Ca^{2+} signaling in Purkinje neurons. *Cell Calcium* 37: 459–466.
- Harvey RJ, Napper RMA (1991) Quantitative studies of the mammalian cerebellum. *Progress in Neurobiology* 36: 437–463.
- Haseloff J, Dormand EL, Brand AH (1999) Live imaging with green fluorescent protein. *Methods in Molecular Biology* 122: 241–259.
- Haseltine EL, Rawlings JB (2002) Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *Journal of Chemical Physics* 117: 6959–6969.
- Häusler S, Maass W (2007) A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. *Cerebral Cortex* 17: 149–162.
- Häusser M (2003) Revealing the properties of dendritic voltage-gated channels: A new approach to the space clamp problem. *Biophysical Journal* 84: 3497–3498.
- Häusser M, Mel B (2003) Dendrites: Bug or feature? *Current Opinion in Neurobiology* 13: 372–383.
- Häusser M, Roth A (1997a) Dendritic and somatic glutamate receptor channels in rat cerebellar Purkinje cells. *Journal of Physiology* 501: 77–95.
- Häusser M, Roth A (1997b) Estimating the time course of the excitatory synaptic conductance in neocortical pyramidal cells using a novel voltage jump method. *Journal of Neuroscience* 17: 7606–7625.
- He L, Wu XS, Mohan R, Wu LG (2006) Two modes of fusion pore opening revealed by cell-attached recordings at a synapse. *Nature* 444: 102–105.
- Hebb DO (1949) *The organization of behavior: A neuropsychological theory*. New York: John Wiley.
- Heine M, Groc L, Frischknecht R, Béïque JC, Lounis B, Rumbaugh G, Huganir RL, Cognet L, Choquet D (2008). Surface mobility of postsynaptic AMPARs tunes synaptic transmission. *Science* 320: 201–205.
- Heinrich R, Jacobs K, Lakes-Harlen R (1998) Tracing of a neuronal network in the locust by pressure injection of markers into a synaptic neuropil. *Journal of Neuroscience Methods* 80: 81–89.
- Helmchen F, Imoto K, Sakmann B (1996) Ca^{2+} buffering and action potential-evoked Ca^{2+} signaling in dendrites of pyramidal neurons. *Biophysical Journal* 70: 1069–1081.
- Hennig MH, Postlethwaite M, Forsythe ID, Graham BP (2008) Interactions between multiple sources of short-term plasticity during evoked and spontaneous activity at the rat calyx of Held. *Journal of Physiology* 586: 3129–3146.
- Hernjak N, Slepchenko BM, Fernald K, Fink CC, Fortin D, Moraru I, Watras J, Loew LM (2005) Modeling and analysis of calcium signaling events leading to long-term depression in cerebellar Purkinje cells. *Biophysical Journal* 89: 3790–3806.
- Hessler D, Young SJ, Carragher BO, Martone ME, Hinshaw JE, Milligan RA, Maslia E, Whittaker M, Lamont S, Ellisman MH (1992) SYNU: Software for visualization of three-dimensional biological structures. *Microscopy* 22: 72–82.
- Hestrin S, Armstrong WE (1996) Morphology and physiology of cortical neurons in layer 1. *Journal of Neuroscience* 16: 5290–5300.
- Higham DJ (2001) An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review* 43: 525–546.
- Hill AV (1910) The possible effects of the aggregation of the molecules of hemoglobin on its dissociation curves. *Journal of Physiology* 40: iv–vii.
- Hill AA, Lu J, Masino MA, Olsen OH, Calabrese RL (2001) A model of a segmental oscillator in the leech heartbeat neuronal network. *Journal of Computational Neuroscience* 10: 281–302.
- Hill AA, Masino MA, Calabrese RL (2002) Model of intersegmental coordination in the leech heartbeat neuronal network. *Journal of Neurophysiology* 87: 1586–1602.
- Hill TL and Chen YD (1972) On the theory of ion transport across nerve membranes. VI. Free energy and activation free energies of conformational change. *Proceedings of the National Academy of Sciences of the USA* 69: 1723–1726.

- Hille B (2001) *Ion Channels of Excitable Membranes*, 3rd ed. Sunderland, MA: Sinauer Associates.
- Hillman DE (1979) Neuronal shape parameters and substructures as a basis of neuronal form. In F Schmitt, ed., *The Neurosciences: Fourth Study Program*, pp. 477–498. Cambridge, MA: MIT Press.
- Hindmarsh JL, Rose RM (1984) A model of neuronal bursting using three coupled first-order differential equations. *Proceedings of the Royal Society London Series B* 221: 87–102.
- Hines M (1984) Efficient computation of branched nerve equations. *International Journal of Biomedical Computing* 15: 69–76.
- Hines ML, Carnevale NT (1997) The NEURON simulation environment. *Neural Computation* 9: 1179–1209. Extended preprint available at <http://www.neuron.yale.edu/neuron/papers/nc97/nctoc.htm>.
- Hines, ML, Carnevale, NT (2001) NEURON: A tool for neuroscientists. *Neuroscientist* 7: 123–135.
- Hines ML, Eichner H, Schürmann F (2008a) Neuron splitting in compute-bound parallel network simulations enables runtime scaling with twice as many processors. *Journal of Computational Neuroscience* 25: 203–210.
- Hines ML, Markram H, Schürmann F (2008b) Fully implicit parallel simulation of single neurons. *Journal of Computational Neuroscience* 25: 439–448.
- Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* 117: 500–544.
- Holland JH (1975) *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.
- Holmes P, Full RJ, Koditschek D, Guckenheimer J (2006) The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review* 48: 207–304.
- Holmes WR (1989) The role of dendritic diameters in maximizing the effectiveness of synaptic inputs. *Brain Research* 478: 127–137.
- Holmes WR (1995) Modeling the effect of glutamate diffusion and uptake on NMDA and non-NMDA receptor saturation. *Biophysical Journal* 69: 1734–1747.
- Holmes WR (2000) Models of calmodulin trapping and CaM kinase II activation in a dendritic spine. *Journal of Computational Neuroscience* 8: 65–85.
- Holmes WR, Grover LM (2006) Quantifying the magnitude of changes in synaptic-level parameters with long-term potentiation. *Journal of Neurophysiology* 96: 1478–1491.
- Holmes WR, Levy WB (1990) Insights into associative long-term potentiation from computational models of NMDA receptor-mediated calcium influx and intracellular calcium concentration changes. *Journal of Neurophysiology* 63: 1148–1168.
- Holmes WR, Rall W (1992a) Electrotonic length estimates in neurons with dendritic tapering or somatic shunt. *Journal of Neurophysiology* 68: 1421–1437.
- Holmes WR, Rall W (1992b) Estimating the electrotonic structure of neurons with compartmental models. *Journal of Neurophysiology* 68: 1438–1452.
- Holmes WR, Woody CD (1989) Effects of uniform and non-uniform synaptic deactivation distributions on the cable properties of modeled cortical pyramidal neurons. *Brain Research* 505: 12–22.
- Holmes WR, Segev I, Rall W (1992) Interpretation of time constant and electrotonic length estimates in multi-cylinder or branched neuronal structures. *Journal of Neurophysiology* 68: 1401–1420.
- Holmes WR, Ambros-Ingerson J, Grover LM (2006) Fitting experimental data to models that use morphological data from public databases. *Journal of Computational Neuroscience* 20: 349–365.
- Holt GR (1998) A critical reexamination of some assumptions and implications of cable theory in neurobiology. Ph. D. thesis, California Institute of Technology, Computation and Neural Systems Program. <http://www-lnc.usc.edu/~holt/papers/thesis/index.html>.
- Horcholle-Bossavit G, Gogoi P, Ivanov Y, Korogod S, Tyc-Dumont S (2000) The problem of the morphological noise in reconstructed dendritic arborizations. *Journal of Neuroscience Methods* 95: 83–93.
- Horikawa H, Armstrong WE (1988) A versatile means of intracellular labeling: Injection of biocytin and its detection with avidin conjugates. *Journal of Neuroscience Methods* 1: 1–11.

- Horn RJ, Patlak J and Stevens CF (1981) Sodium channels need not open before they inactivate. *Nature* 291: 426–427.
- Hosoya T, Baccus SA, Meister M (2005) Dynamic predictive coding by the retina. *Nature* 436: 71–77.
- Howell FW, Dyhrfjeld-Johnsen J, Maex R, Goddard N, De Schutter E (2000) A large-scale model of the cerebellar cortex using pGenesis. *Neurocomputing* 32–33: 1041–1046.
- Huguenard JR, McCormick DA (1992) Simulation of the currents involved in rhythmic oscillations in thalamic relay neurons. *Journal of Neurophysiology* 68: 1373–1383.
- Huguenard JR, Prince DA (1992) A novel T-type current underlies prolonged calcium-dependent burst firing in GABAergic neurons of rat thalamic reticular nucleus. *Journal of Neuroscience* 12: 3804–3817.
- Huntsman MM, Huguenard JR (2000) Nucleus-specific differences in GABA(A)-receptor-mediated inhibition are enhanced during thalamic development. *Journal of Neurophysiology* 83: 350–358.
- Hutcheon B, Yarom Y (2000) Resonance, oscillation and the intrinsic frequency preferences of neurons. *Trends in Neurosciences* 23: 216–222.
- Huys QJ, Ahrens MB, Paninski L (2006) Efficient estimation of detailed single-neuron models. *Journal of Neurophysiology* 96: 872–890.
- Iansek R, Redman SJ (1973) The amplitude, time-course and charge of unitary excitatory post-synaptic potentials evoked in spinal motoneurone dendrites. *Journal of Physiology* 234: 665–688.
- Ichikawa K (2005) A modeling environment with three-dimensional morphology, A-cell-3D, and Ca^{2+} dynamics in a spine. *Neuroinformatics* 3: 49–64.
- Idiart MAP, Abbott LF (1993) Propagation of excitation in neural network models. *Network: Computation in Neural Systems* 4: 285–294.
- Innocenti GM, Lehmann P, Houzel JC (1994) Computational structure of visual callosal axons. *European Journal of Neuroscience* 6: 918–935.
- Isaac JT, Nicoll RA, Malenka RC (1995) Evidence for silent synapses: Implications for the expression of LTP. *Neuron* 15: 427–434.
- Ishizuka N, Cowan WM, Amaral DG (1995) A quantitative analysis of the dendritic organization of pyramidal cells in the rat hippocampus. *Journal of Comparative Neurology* 362: 17–45.
- Ito K, Sass H, Urban J, Hofbauer A, Schneuly S (1997) GAL4-responsive UAS-tau as a tool for studying the anatomy and development of the *Drosophila* nervous system. *Cell and Tissue Research* 290: 1–10.
- Ivanov AI, Calabrese RL (2000) Intracellular Ca^{2+} dynamics during spontaneous and evoked activity of leech heart interneurons: Low-threshold Ca currents and graded synaptic transmission. *Journal of Neuroscience* 20: 4930–4943.
- Ivanov AI, Calabrese RL (2003) Modulation of spike-mediated synaptic transmission by presynaptic background Ca^{2+} in leech heart interneurons. *Journal of Neuroscience* 23: 1206–1218.
- Iwasaki T, Zheng M (2006) Sensory feedback mechanism underlying entrainment of central pattern generator to mechanical resonance. *Biological Cybernetics* 94: 245–261.
- Izhikevich EM (2001) Resonate-and-fire neurons. *Neural Networks* 14: 883–894.
- Izhikevich EM (2003) Simple model of spiking neurons. *IEEE Transactions on Neural Networks* 14: 1569–1572.
- Izhikevich EM (2004) Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks* 15: 1063–1070.
- Izhikevich E (2006) *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, MA: MIT Press.
- Izhikevich EM, Gally JA, Edelman GM (2004) Spike-timing dynamics of neuronal groups. *Cerebral Cortex* 14: 933–944.
- Jack JJB, Noble D, Tsien RW (1975) *Electric Current Flow in Excitable Cells*. Oxford: Clarendon Press.
- Jacobs GA, Nevin R (1991) Anatomical relationships between sensory afferent arborizations in the cricket cercal system. *Anatomical Record* 231: 563–572.
- Jacobs GA, Theunissen FE (1996) Functional organization of a neural map in the cricket cercal sensory system. *Journal of Neuroscience* 16: 769–784.

- Jacobs GA, Theunissen FE (2000) Extraction of sensory parameters from a neural map by primary sensory interneurons. *Journal of Neuroscience* 20: 2934–2943.
- Jaeger D (2001) Accurate reconstruction of neuronal morphology. In E De Schutter, ed., *Computational Neuroscience: Realistic Modeling for Experimentalists*, pp. 159–178. Boca Raton, FL: CRC Press.
- Jaeger D, Bower JM (1999) Synaptic control of spiking in cerebellar Purkinje cells: Dynamic current clamp based on model conductances. *Journal of Neuroscience* 19: 6090–6101.
- Jaeger D, De Schutter E, Bower JM (1997) The role of synaptic and voltage-gated currents in the control of Purkinje cell spiking: A modeling study. *Journal of Neuroscience* 17: 91–106.
- Jahnsen H, Llinás RR (1984) Ionic basis for the electroresponsiveness and oscillatory properties of guinea-pig thalamic neurons in vitro. *Journal of Physiology* 349: 227–247.
- Jahr CE, Stevens CF (1990) A quantitative description of NMDA receptor-channel kinetic behavior. *Journal of Neuroscience* 10: 1830–1837.
- Jain R, Kasturi R, Schunk B (1995) *Machine Vision*. New York: McGraw Hill.
- Jefferys JGR (1995) Nonsynaptic modulation of neuronal activity in the brain: Electric currents and extracellular ions. *Physiological Reviews* 75: 689–723.
- Jensen FE, Harris KM (1989) Preservation of neuronal ultrastructure in hippocampal slices using rapid microwave-enhanced fixation. *Journal of Neuroscience Methods* 29: 217–230.
- Jerng HH, Pfaffinger PJ, Covarrubias M (2004) Molecular physiology and modulation of somatodendritic A-type potassium channels. *Molecular and Cellular Neuroscience* 27: 343–369.
- Jezzini SH, Hill AA, Kuzyk P, Calabrese RL (2004) Detailed model of intersegmental coordination in the timing network of the leech heartbeat central pattern generator. *Journal of Neurophysiology* 91: 958–977.
- Johnson DH (1996) Point process models of single-neuron discharges. *Journal of Computational Neuroscience* 3: 275–299.
- Johnson FH, Eyring H, Stover BJ (1974) *The Theory of Rate Processes in Biology and Medicine*. New York: Wiley.
- Johnson SE, Murphrey RK (1985) The afferent projection of mesothoracic bristle hairs in the cricket *Acheta domesticus*. *Journal of Comparative Physiology* 156: 369–379.
- Johnston D (1981) Passive cable properties of hippocampal CA3 pyramidal neurons. *Cellular and Molecular Neurobiology* 1: 41–55.
- Johnston D, Brown TH (1983) Interpretation of voltage-clamp measurements in hippocampal neurons. *Journal of Neurophysiology* 50: 464–486.
- Johnston D, Wu SM (1995) *Foundations of Cellular Neurophysiology*. Cambridge, MA: MIT Press.
- Jolivet R, Rauch A, Lüscher HR, Gerstner W (2006) Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of Computational Neuroscience* 21: 35–49.
- Jonas P (2000) The time course of signaling at central glutamatergic synapses. *News of Physiological Sciences* 15: 83–89.
- Jonas P, Spruston N (1994) Mechanisms shaping glutamate-mediated excitatory postsynaptic currents in the CNS. *Current Opinion in Neurobiology* 4: 366–372.
- Jones SP, Rahimi O, O’Boyle MP, Diaz DL, Claiborne BJ (2003) Maturation of granule cell dendrites after mossy fiber arrival in hippocampal field CA3. *Hippocampus* 13: 413–437.
- Joshi I, Shokralla S, Titis P, Wang LY (2004) The role of AMPA receptor gating in the development of high-fidelity neurotransmission at the calyx of Held synapse. *Journal of Neuroscience* 24: 183–196.
- Kampa BM, Clements J, Jonas P, Stuart GJ (2004) Kinetics of Mg^{2+} unblock of NMDA receptors: Implications for spike-timing dependent synaptic plasticity. *Journal of Physiology* 556: 337–345.
- Karameh F, Massaquoi S (2005) A model of nonlinear motor cortical integration and its relation to movement speed profile control. In *27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* vol. 4, pp. 4331–4336. Piscataway, NJ: IEEE Press.
- Karameh FN, Dahleh MA, Brown EN, Massaquoi SG (2006) Modeling the contribution of lamina 5 neuronal and network dynamics to low-frequency EEG phenomena. *Biological Cybernetics* 95: 289–310.

- Kasai H, Petersen OH (1994) Spatial dynamics of second messengers: IP₃ and cAMP as long-range and associative messengers. *Trends in Neurosciences* 17: 95–101.
- Kay AR (1992) An intracellular medium formulary. *Journal of Neuroscience Methods* 44: 91–100.
- Keizer J, Levine L (1996) Ryanodine receptor adaptation and Ca²⁺-induced Ca²⁺ release-dependent Ca²⁺ oscillations. *Biophysical Journal* 71: 3477–3487.
- Keller DX, Franks KM, Bartol TM, Sejnowski TJ (2008) Calmodulin activation by calcium transients in the postsynaptic density of dendritic spines. *PLoS ONE* 3: e2045.
- Kempter R, Leibold C, Wagner H, van Hemmen JL (2001) Formation of temporal-feature maps by axonal propagation of synaptic learning. *Proceedings of the National Academy of Sciences of the USA* 98: 4166–4171.
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1942–1948. Piscataway, NJ: IEEE Press.
- Kenyon GT, Medina JF, Mauk MD (1998) A mathematical model of the cerebellar-olivary system I: Self-regulating equilibrium of climbing fiber activity. *Journal of Computational Neuroscience* 5: 17–33.
- Keren N, Peled N, Korngreen A (2005) Constraining compartmental models using multiple voltage recordings and genetic algorithms. *Journal of Neurophysiology* 94: 3710–3742.
- Khalil ZM, Gouwens NW, Raman IM (2003) The contribution of resurgent sodium current to high-frequency firing in Purkinje neurons: An experimental and modeling study. *Journal of Neuroscience* 23: 4899–4912.
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220: 671–680.
- Kirov SA, Harris KM (1999) Dendrites are more spiny on mature hippocampal neurons when synapses are inactivated. *Nature Neuroscience* 2: 878–883.
- Kirov SA, Petrak LJ, Fiala JC, Harris KM (2004) Dendritic spines disappear with chilling but proliferate excessively upon rewarming of mature hippocampus. *Neuroscience* 127: 69–80.
- Kleinfeld D, Sompolinsky H (1988) Associative neural network model for the generation of temporal patterns. Theory and application to central pattern generators. *Biophysical Journal* 54: 1039–1051.
- Kliemann W (1987) A stochastic dynamical model for the characterization of the geometrical structure of dendritic processes. *Bulletin of Mathematical Biology* 49: 135–152.
- Kloeden PE, Platen E (2000) *Numerical Solution of Stochastic Differential Equations*, 1st ed. Berlin: Springer-Verlag.
- Knight BW (1972) Dynamics of encoding in a population of neurons. *Journal of General Physiology* 59: 734–766.
- Koch C (1999) *Biophysics of Computation: Information Processing in Single Neurons*. New York: Oxford University Press.
- Koch C, Poggio T, Torre V (1983) Nonlinear interactions in the dendritic tree: Localization, timing and role in information processing. *Proceedings of the National Academy of Sciences of the USA* 80: 2799–2802.
- Kole MH, Ilshner SU, Kampa BM, Williams SR, Ruben PC, Stuart GJ (2008) Action potential generation requires a high sodium channel density in the axon initial segment. *Nature Neuroscience* 11: 178–186.
- Kopysova IL, Debanne D (1998) Critical role of axonal A-type K⁺ channels and axonal geometry in the gating of action potential propagation along CA3 pyramidal cell axons: A simulation study. *Journal of Neuroscience* 18: 7436–7451.
- Koslow, SH, Hirsch, MD (2004) Celebrating a decade of neuroscience databases: Looking to the future of high-throughput data analysis, data integration and discovery neuroscience. *Neuroinformatics* 2: 267–270.
- Kotaleski JH, Plenz D, Blackwell KT (2006) Using potassium currents to solve signal-to-noise problems in inhibitory feedforward networks of the striatum. *Journal of Neurophysiology* 95: 331–341.
- Kötter R (2004) Online retrieval, processing, and visualization of primate connectivity data from the CoCoMac database. *Neuroinformatics* 2: 127–144.
- Kraushaar U, Jonas P (2000). Efficacy and stability of quantal GABA release at a hippocampal interneuron-principal neuron synapse. *Journal of Neuroscience* 20: 5594–5607.

- Krichmar JL, Nasuto SJ, Scorcioni R, Washington SD, Ascoli GA (2002) Effects of dendritic morphology on CA3 pyramidal cell electrophysiology: A simulation study. *Brain Research* 941: 11–28.
- Kriener B, Tetzlaff T, Aertsen A, Diesmann M, Rotter S (2008) Correlations and population dynamics in cortical networks. *Neural Computation* 20: 2185–2226.
- Krink T, Filipic B, Fogel GB (2004) Noisy optimization problems—a particular challenge for differential evolution? In *Proceedings of 2004 Congress on Evolutionary Computation*, pp. 332–339. Piscataway, NJ: IEEE press.
- Kristan WB, Jr., Calabrese RL, Friesen WO (2005) Neuronal control of leech behavior. *Progress in Neurobiology* 76: 279–327.
- Kubota Y, Putkey JA, Shouval HZ, Waxham MN (2008) IQ-motif proteins influence intracellular free Ca²⁺ in hippocampal neurons through their interactions with calmodulin. *Journal of Neurophysiology* 99: 264–276.
- Kullmann PHM, Wheeler DW, Beacom J, Horn JP (2004) Implementation of a fast 16-bit dynamic clamp using LabVIEW-RT. *Journal of Neurophysiology* 91: 542–554.
- Kumar A, Rotter S, Aertsen A (2008) Conditions for propagating synchronous spiking and asynchronous firing rates in a cortical network model. *Journal of Neuroscience* 28: 5268–5280.
- Kuo PD, Eliasmith C (2005) Integrating behavioral and neural data in a model of zebrafish network interaction. *Biological Cybernetics* 93: 178–187.
- Labarca P, Rice JA, Fredkin DR, Montal M (1985) Kinetic analysis of channel gating. Application to the cholinergic receptor channel and the chloride channel from *Torpedo Californica*. *Biophysical Journal* 47: 469–478.
- Lapicque L (1907) Recherches quantitatives sur l'excitabilité électrique des nerfs traitée comme une polarisation. *Journale de Physiologie et Pathologie Générale* 9: 620–635.
- Larkman A, Mason A (1990) Correlations between morphology and electrophysiology of pyramidal neurons in slices of rat visual cortex. I. Establishment of cell classes. *Journal of Neuroscience* 10: 1407–1414.
- Larkman AU, Major G, Stratford KJ, Jack JJB (1992) Dendritic morphology of pyramidal neurons of the visual cortex of the rat. IV: Electrical geometry. *Journal of Comparative Neurology* 323: 137–152.
- Larkum ME, Senn W, Luscher H-R (2004) Top-down dendritic input increases the gain of layer 5 pyramidal neurons. *Cerebral Cortex* 14: 1059–1070.
- Laughlin SB, Sejnowski TJ (2003) Communication in neuronal networks. *Science* 301: 1870–1874.
- LeMasson G, Maex R (2001) Introduction to equation solving and parameter fitting. In E De Schutter, ed., *Computational Neuroscience: Realistic Modeling for Experimentalists*, pp. 1–24. Boca Raton, FL: CRC Press.
- LeMasson G, LeMasson S, Moullins M (1995) From conductances to neural network properties: Analysis of simple circuits using the hybrid network method. *Progress in Biophysics and Molecular Biology* 64: 201–220.
- Le Van Quyen M, Bragin A (2007) Analysis of dynamic brain oscillations: Methodological advances. *Trends in Neurosciences* 30: 365–373.
- Lebedev MA, Nicolelis MA (2006) Brain-machine interfaces: Past, present and future. *Trends in Neurosciences* 29: 536–546.
- LeBlanc J, Ward MO, Wittels N (1990) Exploring n -dimensional databases. In A Kaufman, ed., *First IEEE Conference on Visualization*, pp. 230–237. Piscataway, NJ: IEEE Press.
- L'Ecuyer P, Simard R (2007) TestU01: A C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software* 33: 22.
- Lee T, Luo L (1999) Mosaic analysis with a repressible cell marker for studies of gene function in neuronal morphogenesis. *Neuron* 22: 451–461.
- Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics* 2: 164–168.
- Levine RB, Murphrey RK (1980) Pre- and postsynaptic inhibition of identified giant interneurons in the cricket (*Acheta domesticus*). *Journal of Comparative Physiology A* 135: 269–282.

- Lewis TJ, Rinzel J (2000) Self-organized synchronous oscillations in a network of excitable cells coupled by gap junctions. *Network: Computation in Neural Systems* 11: 299–320.
- Lewis TJ, Rinzel J (2003) Dynamics of spiking neurons connected by both inhibitory and electrical coupling. *Journal of Computational Neuroscience* 14: 283–309.
- Lewit-Bentley A, Rety S (2000) EF-hand calcium-binding proteins. *Current Opinion in Structural Biology* 10: 637–643.
- Li Y-, Rinzel J (1994) Equations for InsP₃ receptor-mediated [Ca²⁺]_i oscillations derived from a detailed kinetic model: A Hodgkin-Huxley-like formalism. *Journal of Theoretical Biology* 166: 461–473.
- Liao D, Hessler NA, Malinow R (1995) Activation of postsynaptically silent synapses during pairing-induced LTP in CA1 region of hippocampal slice. *Nature* 375: 400–404.
- Lindner B, Schimansky-Geier L (2001) Transmission of noise-coded versus additive signals through a neuronal ensemble. *Physical Review Letters* 86: 2934–2937.
- Lindner B, Longtin A, Bulsara A (2003) Analytic expressions for rate and CV of a type I neuron driven by white Gaussian noise. *Neural Computation* 15: 1760–1787.
- Lisman J, Raghavachari S (2006) A unified model of the presynaptic and postsynaptic changes during LTP at CA1 synapses. *Science STKE*: re11.
- Lisman J, Schulman H, Cline H (2002) The molecular basis of CaMKII function in synaptic and behavioral memory. *Nature Reviews Neuroscience* 3: 175–190.
- Lisman JE, Raghavachari S, Tsien RW (2007). The sequence of events that underlie quantal transmission at central glutamatergic synapses. *Nature Reviews Neuroscience* 8: 597–609.
- Liu Z, Golowasch J, Marder E, Abbott LF (1998) A model neuron with activity-dependent conductances regulated by multiple calcium sensors. *Journal of Neuroscience* 18: 2309–2320.
- Llinás RR (1988) The intrinsic electrophysiological properties of mammalian neurons: A new insight into CNS function. *Science* 242: 1654–1664.
- Llinás RR, Grace AA, Yarom Y (1991) In vitro neurons in mammalian cortical layer 4 exhibit intrinsic oscillatory activity in the 10- to 50-Hz frequency range. *Proceedings of the National Academy of Sciences of the USA* 88: 897–901. Erratum in *Proceedings of the National Academy of Sciences of the USA* 88: 3510.
- London M, Meunier C, Segev I (1999) Signal transfer in passive dendrites with nonuniform membrane conductance. *Journal of Neuroscience* 19: 8219–8233.
- Longobardo G, Evangelisti CJ, Cherniack NS (2005) Introduction of respiratory pattern generators into models of respiratory control. *Respiratory Physiology and Neurobiology* 148: 285–301.
- Lopreore CL, Bartol TM, Coggan JS, Keller DX, Sosinsky GE, Ellisman MH, Sejnowski TJ (2008) Computational modeling of three-dimensional electrodiffusion in biological systems: Application to the node of Ranvier. *Biophysical Journal* 95: 2624–2635.
- Losonczy A, Magee JC (2006) Integrative properties of radial oblique dendrites in hippocampal CA1 pyramidal neurons. *Neuron* 50: 291–307.
- Luczak A (2006) Spatial embedding of neuronal trees modeled by diffusive growth. *Journal of Neuroscience Methods* 157: 132–141.
- Lux H-D, Schubert P, Kreutzberg GW (1970) Direct matching of morphological and electrophysiological data in cat spinal motoneurones. In P Andersen, JKS Jansen, eds., *Excitatory Synaptic Mechanisms*, pp. 189–198. Oslo: Universitetsforlaget.
- Lytton WW (1996). Optimizing synaptic conductance calculation for network simulations. *Neural Computation* 8: 501–509.
- Lytton WW, Hines ML (2005) Independent variable time-step integration of individual neurons for network simulations. *Neural Computation* 17: 903–921.
- Maass W, Zador AM (1999) Dynamic stochastic synapses as computational units. *Neural Computation* 11: 903–917.
- Macdonald M (1989) *Biological Delay Systems: Linear Stability Theory*. Cambridge, UK: Cambridge University Press.
- Madden DR (2005) New light on an open-and-shut case. *Nature Chemical Biology* 1: 317–319.

- Maex R, De Schutter E (1998) Synchronization of Golgi and granule cell firing in a detailed network model of the cerebellar granule cell layer. *Journal of Neurophysiology* 80: 2521–2537.
- Maex R, De Schutter E (2003) Resonant synchronization in heterogeneous networks of inhibitory neurons. *Journal of Neuroscience* 23: 10503–10514.
- Maex R, De Schutter E (2005) Oscillations in the cerebellar cortex: A computational analysis of their frequency bands. *Progress in Brain Research* 148: 181–188.
- Maex R, De Schutter E (2007) Mechanism of spontaneous and self-sustained oscillations in networks coupled through axo-axonal gap junctions. *European Journal of Neuroscience* 25: 3347–3358.
- Maex R, Orban GA (1996) Model circuit of spiking neurons generating directional selectivity in simple cells. *Journal of Neurophysiology* 75: 1515–1545.
- Maex R, Vos BP, De Schutter E (2000) Weak common parallel fiber synapses explain the loose synchrony observed between rat cerebellar Golgi cells. *Journal of Physiology* 523: 175–192.
- Mainen ZF, Sejnowski TJ (1995) Reliability of spike timing in neocortical neurons. *Science* 268: 1503–1506.
- Mainen ZF, Sejnowski TJ (1996) Influence of dendritic structure on firing patterns in model neocortical neurons. *Nature* 382: 363–366.
- Mainen ZF, Joerges J, Huguenard JR, Sejnowski TJ (1995) A model of spike initiation in neocortical pyramidal neurons. *Neuron* 15: 1427–1439.
- Mainen ZF, Carnevale NT, Zador AM, Claiborne BJ, Brown TH (1996) Electrotonic architecture of hippocampal CA1 pyramidal neurons based on three-dimensional reconstructions. *Journal of Neurophysiology* 76: 1904–1923.
- Major G, Evans JD, Jack JJB (1993a) Solutions for transients in arbitrarily branching cables: I. Voltage recording with a somatic shunt. *Biophysical Journal* 65: 423–449.
- Major G, Evans JD, Jack JJB (1993b) Solutions for transients in arbitrarily branching cables: II. Voltage clamp theory. *Biophysical Journal* 65: 450–568.
- Major G, Larkman AU, Jonas P, Sakmann B, Jack JJB (1994) Detailed passive cable models of whole-cell recorded CA3 pyramidal neurons in rat hippocampal slices. *Journal of Neuroscience* 14: 4613–4638.
- Makarov VA, Nekorkin VI, Velarde MG (2001) Spiking behavior in a noise-driven system combining oscillatory and excitatory properties. *Physical Review Letters* 86: 3431–3434.
- Malenfort MG, Hamm TM (2004) Estimation of the electrical parameters of spinal motoneurons using impedance measurements. *Journal of Neurophysiology* 92: 1433–1444.
- Mamiya A, Manor Y, Nadim F (2003) Short-term dynamics of a mixed chemical and electrical synapse in a rhythmic network. *Journal of Neuroscience* 23: 9557–9564.
- Manor Y, Nadim F (2001) Synaptic depression mediates bistability in neuronal networks with recurrent inhibitory connectivity. *Journal of Neuroscience* 21: 9460–9470.
- Manor Y, Gonczarowski J, Segev I (1991) Propagation of action-potentials along complex axonal trees. Model and implementation. *Biophysical Journal* 60: 1411–1423.
- Manor Y, Rinzel J, Segev I, Yarom Y (1997) Low-amplitude oscillations in the inferior olive: A model based on electrical coupling of neurons with heterogeneous channel densities. *Journal of Neurophysiology* 77: 2736–2752.
- Manor Y, Yarom Y, Choref E, Devor A (2000) To beat or not to beat: A decision taken at the network level. *Journal of Physiology Paris* 94: 375–390.
- Maravall M, Mainen ZF, Sabatini BL, Svoboda K (2000) Estimating intracellular calcium concentrations and buffering without wavelength ratioing. *Biophysical Journal* 78: 2655–2667.
- Marchetti C, Tabak J, Chub N, O'Donovan MJ, Rinzel J (2005) Modeling spontaneous activity in the developing spinal cord using activity-dependent variations of intracellular chloride. *Journal of Neuroscience* 25: 3601–3612.
- Marder E, Goaillard JM (2006) Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience* 7: 563–574.
- Marder E, Bucher D, Schulz DJ, Taylor AL (2005) Invertebrate central pattern generation moves along. *Current Biology* 15: R685–R699.

- Markram H (2006) The blue brain project. *Nature Reviews Neuroscience* 7: 153–160.
- Markram H, Lübke J, Frotscher M, Roth A, Sakmann B (1997) Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *Journal of Physiology* 500: 409–440.
- Markram H, Wang Y, Tsodyks M (1998) Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences of the USA* 95: 5323–5328.
- Markram H, Toledo-Rodriguez M, Wang Y, Gupta A, Silberberg G, Wu C (2004) Interneurons of the neocortical inhibitory system. *Nature Reviews Neuroscience* 5: 793–807.
- Marmarelis VZ (2004) *Nonlinear Dynamic Modeling of Physiological Systems*. New York: Wiley.
- Marom S, Abbott LF (1994) Modeling state-dependent inactivation of membrane currents. *Biophysical Journal* 67: 515–520.
- Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics* 11: 431–441.
- Marsaglia G (2003) Diehard battery of tests of randomness v0.2 beta, <http://www.cs.hku.hk/~diehard/>.
- Martone ME, Gupta A, Wong M, Qian X, Sosinsky G, Ludascher B, Ellisman MH (2002) A cell-centered database for electron tomographic data. *Journal of Structural Biology* 138: 145–155.
- Martone ME, Zhang S, Gupta A, Qian X, He H, Price D, Wong M, Santini S, Ellisman MH (2003) The cell-centered database: A database for multiscale structural and protein localization data from light and electron microscopy. *Neuroinformatics* 1: 379–396.
- Mascagni MV, Sherman AS (1998) Numerical methods for neuronal modeling. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 569–606. Cambridge, MA: MIT Press.
- Masino MA, Calabrese RL (2002a) A functional asymmetry in the leech heartbeat timing network is revealed by driving the network across various cycle periods. *Journal of Neuroscience* 22: 4418–4427.
- Masino MA, Calabrese RL (2002b) Phase relationships between segmentally organized oscillators in the leech heartbeat pattern generating network. *Journal of Neurophysiology* 87: 1572–1585.
- Masino MA, Calabrese RL (2002c) Period differences between segmental oscillators produce intersegmental phase differences in the leech heartbeat timing network. *Journal of Neurophysiology* 87: 1603–1615.
- Mason A, Larkman A (1990) Correlations between morphology and electrophysiology of pyramidal neurons in slices of rat visual cortex. II. Electrophysiology. *Journal of Neuroscience* 10: 1415–1428.
- Matsumoto M, Nishimura T (1998) Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 8: 3–30.
- Matveev V, Sherman A, Zucker RS (2002) New and corrected simulations of synaptic facilitation. *Biophysical Journal* 83: 1368–1373.
- Mayer ML (2005) Crystal structures of the GLUR5 and GLUR6 ligand binding cores: Molecular mechanisms underlying kainate receptor selectivity. *Neuron* 45: 539–552.
- Mayer ML (2006) Glutamate receptors at atomic resolution. *Nature* 440: 456–462.
- Mayer ML, Armstrong N (2004) Structure and function of glutamate receptor ion channels. *Annual Review of Physiology* 66: 161–181.
- Mayer ML, Westbrook GL (1987) Permeation and block of *N*-methyl-D-aspartic acid receptor channels by divalent cations in mouse cultured central neurones. *Journal of Physiology* 394: 501–527.
- McCormick DA, Huguenard JR (1992) A model of the electrophysiological properties of thalamocortical relay neurons. *Journal of Neurophysiology* 68: 1384–1400.
- McCormick DA, Wang Z, Huguenard J (1993) Neurotransmitter control of neocortical neuronal activity and excitability. *Cerebral Cortex* 3: 387–398.
- McCormick DA, Shu Y, Yu Y (2007) Neurophysiology: Hodgkin and Huxley model—still standing? *Nature* 445: 1–2.
- McDonald AJ (1992) Neuroanatomical labeling with biocytin: A review. *Neuroreport* 10: 821–827.
- McHugh JM, Kenyon JL (2004) An Excel-based model of Ca^{2+} diffusion and fura 2 measurements in a spherical cell. *American Journal of Physiology—Cell Physiology* 286: C342–C348.

- McIntyre CC, Richardson AG, Grill WM (2002) Modeling the excitability of mammalian nerve fibers: Influence of afterpotentials on the recovery cycle. *Journal of Neurophysiology* 87: 995–1006.
- McIntyre CC, Grill WM, Sherman DL, Thakor NV (2004) Cellular effects of deep brain stimulation: Model-based analysis of activation and inhibition. *Journal of Neurophysiology* 91: 1457–1469.
- Means S, Smith AJ, Shepherd J, Shadid J, Fowler J, Wojcikiewicz RJ, Mazel T, Smith GD, Wilson BS (2006) Reaction diffusion modeling of calcium dynamics with realistic ER geometry. *Biophysical Journal* 91: 537–557.
- Medina JF, Mauk MD (1999) Simulations of cerebellar motor learning: Computational analysis of plasticity at the mossy fiber to deep nucleus synapse. *Journal of Neuroscience* 19: 7140–7151.
- Megías, M, Emri, ZS, Freund, TF, Gulyás, AI (2001) Total number and distribution of inhibitory and excitatory synapses on hippocampal CA1 pyramidal cells. *Neuroscience* 103: 527–540.
- Meinrenken CJ, Borst JGG, Sakmann B (2002) Calcium secretion coupling at calyx of Held governed by nonuniform channel-vesicle topography. *Journal of Neuroscience* 22: 1648–1667.
- Meir A, Ginsburg S, Butkevich A, Kachalsky SG, Kaiserman I, Ahdut R, Demirgoren S, Rahamimoff R (1999) Ion channels in presynaptic nerve terminals and control of transmitter release. *Physiological Reviews* 79: 1019–1088.
- Mennerick S, Que J, Benz A, Zorumski CF (1995) Passive and synaptic properties of hippocampal neurons grown in microcultures and in mass cultures. *Journal of Neurophysiology* 73: 320–332.
- Merriam EB, Netoff TI, Banks MI (2005) Bistable network behavior of layer interneurons in auditory cortex. *Journal of Neuroscience* 25: 6175–6186.
- Meunier C, Borejsza K (2005) How membrane properties shape the discharge of motoneurons: A detailed analytical study. *Neural Computation* 17: 2383–2420.
- Michaelis L, Menten ML (1913) Die Kinetik der invertinwirkung. *Biochemische Zeitschrift* 49: 333–349.
- Migliore M, Cook EP, Jaffe DB, Turner DA, Johnston D (1995) Computer simulations of morphologically reconstructed CA3 hippocampal neurons. *Journal of Neurophysiology* 73: 1157–1168.
- Migliore M, Hoffman DA, Magee JC, Johnston D (1999) Role of an A-type K^+ conductance in the back-propagation of action potentials in the dendrites of hippocampal pyramidal neurons. *Journal of Computational Neuroscience* 7: 5–15.
- Migliore M, Ferrante M, Ascoli GA (2005a) Signal propagation in oblique dendrites of CA1 pyramidal cells. *Journal of Neurophysiology* 94: 4145–4155.
- Migliore M, Hines ML, Shepherd GM (2005b) The role of distal dendritic gap junctions in synchronization of mitral cell axonal output. *Journal of Computational Neuroscience* 18: 151–161.
- Migliore M, Cannia C, Lytton WW, Markram H, Hines ML (2006) Parallel network simulations with NEURON. *Journal of Computational Neuroscience* 21: 119–129.
- Miikkulainen R, Bednar JA, Choe YC, Sirosh J (2005) *Computational Maps in the Visual Cortex*. Berlin: Springer-Verlag.
- Miller KD (1996) Synaptic economics: Competition and cooperation in synaptic plasticity. *Neuron* 17: 371–374.
- Miller RJ (1998) Presynaptic receptors. *Annual Review of Pharmacology and Toxicology* 38: 201–227.
- Mino H, Rubinstein J, White J (2002) Comparison of algorithms for the simulation of action potentials with stochastic sodium channels. *Annals of Biomedical Engineering* 30: 578–587.
- Mira, J, Álvarez, JR (eds.) (2003) *An Exponential-Decay Synapse Integrated Circuit for Bio-inspired Neural Networks*, Vol. 2686 of Lecture Notes in Computer Science. Berlin: Springer-Verlag.
- Misra J (1986) Distributed discrete-event simulation. *ACM Computing Surveys* 18: 39–65.
- Miura K, Tsubo Y, Okada M, Fukai T (2007) Balanced excitatory and inhibitory inputs to cortical neurons decouple firing irregularity from rate modulations. *Journal of Neuroscience* 27: 13802–13812.
- Mizrahi, A, Katz, LC (2003) Dendritic stability in the adult olfactory bulb. *Nature Neuroscience* 6: 1201–1207.
- Moczydlowski E, Latorre R (1983) Gating kinetics of Ca^{2+} -activated K^+ channels from rat muscle incorporated into planar lipid bilayers. Evidence for two voltage-dependent Ca^{2+} binding reactions. *Journal of General Physiology* 82: 511–542.

- Mogami H, Nakano K, Tepikin AV, Petersen OH (1997) Ca^{2+} flow via tunnels in polarized cells: Recharging of apical Ca^{2+} stores by focal entry through basal membrane patch. *Cell* 88: 49–55.
- Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genetical Research* 13: 2467–2474.
- Mongillo G, Barak O, Tsodyks M (2008) Synaptic theory of working memory. *Science* 319: 1543–1546.
- Moore GP, Segundo JP, Perkel DH, Levitan H (1970) Statistical signs of synaptic interaction in neurons. *Biophysical Journal* 10: 876–900.
- Moore JW, Ramon F (1974) On numerical integration of the Hodgkin and Huxley equations for a membrane action potential. *Journal of Theoretical Biology* 45: 249–273.
- Morris C, Lecar H (1981) Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal* 35: 193–213.
- Morrison A, Mehring C, Geisel T, Aertsen AD, Diesmann M (2005) Advancing the boundaries of high-connectivity network simulation with distributed computing. *Neural Computation* 17: 1776–1801.
- Morrison A, Aertsen A, Diesmann M (2007) Spike-timing-dependent plasticity in balanced random networks. *Neural Computation* 19: 1437–1467.
- Morrison A, Diesmann M, Gerstner W (2008) Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics* 98: 459–478.
- Morton-Firth CJ, Bray D (1998) Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology* 192: 117–128.
- Mouchet P, Yelnick J (2004) Basic electrotonic properties of primate pallidal neurons as inferred from a detailed analysis of their morphology: A modeling study. *Synapse* 54: 11–23.
- Munsky B, Khammash M (2006) The finite-state projection algorithm for the solution of the chemical master equation. *Journal of Chemical Physics* 124: 44104.
- Nadim F, Golowasch J (2006) Signal transmission between gap-junctionally coupled passive cables is most effective at an optimal diameter. *Journal of Neurophysiology* 95: 3831–3843.
- Nadim F, Manor Y (2000) The role of short-term synaptic dynamics in motor control. *Current Opinion in Neurobiology* 10: 683–690.
- Nadim F, Manor Y, Nusbaum MP, Marder E (1998) Frequency regulation of a slow rhythm by a fast periodic input. *Journal of Neuroscience* 18: 5053–5067.
- Nagumo JS, Arimoto S, Yochizawa S (1962) An active pulse transmission like simulating nerve axon. *Proceedings of the IRE* 50: 2061–2070.
- Naoki H, Sakumura Y, Ishii S (2005) Local signaling with molecular diffusion as a decoder of Ca^{2+} signals in synaptic plasticity. *Molecular Systems Biology* 1: 0027.
- Naraghi M, Neher E (1997) Linearized buffered Ca^{2+} diffusion in microdomains and its implications for calculation of $[\text{Ca}^{2+}]$ at the mouth of a calcium channel. *Journal of Neuroscience* 17: 6961–6973.
- Naud R, Marcille N, Clopath C, Gerstner W (2008) Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics* 99: 335–347.
- Naundorf B, Geisel T, Wolf F (2003) How fast can a neuron react to transient stimuli? Preprint available at, xxx.lanl.gov/physics/0307135.
- Naundorf B, Geisel T, Wolf F (2005) Action potential onset dynamics and the response speed of neuronal populations. *Journal of Computational Neuroscience* 18: 297–309.
- Naundorf B, Wolf F, Volgushev M (2006) Unique features of action potential initiation in cortical neurons. *Nature* 440: 1060–1063.
- Neher E (1998) Usefulness and limitations of linear approximations to the understanding of Ca^{2+} signals. *Cell Calcium* 24: 345–357.
- Nelder JA, Mead R (1965) A simplex method for function minimization. *Computer Journal* 7: 308–313.
- Nelson PG, Lux HD (1970) Some electrical measurements of motoneuron parameters. *Biophysical Journal* 10: 55–73.
- Nenadic Z, Ghosh BK, Ulinski P (2003) Propagating waves in visual cortex: A large-scale model of turtle visual cortex. *Journal of Computational Neuroscience* 14: 161–184.

- Netoff TI, Banks MI, Dorval AD, Acker CD, Haas JS, Kopell N, White JA (2005) Synchronization in hybrid neuronal networks of the hippocampal formation. *Journal of Neurophysiology* 93: 1197–1208.
- Nevian T, Sakmann B (2006) Spine Ca^{2+} signaling in spike-timing-dependent plasticity. *Journal of Neuroscience* 26: 11001–11013.
- Nevian T, Larkum ME, Polsky A, Schiller J (2007). Properties of basal dendrites of layer 5 pyramidal neurons: A direct patch-clamp recording study. *Nature Neuroscience* 10: 206–214.
- Nielsen TA, DiGregorio DA, Silver RA (2004) Modulation of glutamate mobility reveals the mechanism underlying slow-rising AMPAR EPSCs and the diffusion coefficient in the synaptic cleft. *Neuron* 42: 757–771.
- Niles WD, Levis RA, Cohen FS (1988) Planar bilayer membranes made from phospholipid monolayers form by a thinning process. *Biophysical Journal* 53: 327–335.
- Nitzan R, Segev I, Yarom Y (1990) Voltage behavior along the irregular dendritic structure of morphologically and physiologically characterized vagal motoneurons in the guinea pig. *Journal of Neurophysiology* 63: 333–346.
- Norris BJ, Weaver AL, Morris LG, Wenning A, Garcia PA, Calabrese RL (2006) A central pattern generator producing alternative outputs: Temporal pattern of premotor activity. *Journal of Neurophysiology* 96: 309–326.
- Nowak LG, Sanchez-Vives MV, McCormick DA (1997) Influence of low and high frequency inputs on spike timing in visual cortical neurons. *Cerebral Cortex* 7: 487–501.
- Nowakowski RS, Hayes NL, Egger MD (1992) Competitive interactions during dendritic growth: A simple stochastic growth algorithm. *Brain Research* 576: 152–156.
- Nowotny T, Szucs A, Pinto RD, Selverston AI (2006) Stdpc: A modern dynamic clamp. *Journal of Neuroscience Methods* 158: 287–299.
- Nowycky MC, Pinter MJ (1993) Time courses of calcium and calcium-bound buffers following calcium influx in a model cell. *Biophysical Journal* 64: 77–91.
- O'Boyle MP, Rahimi O, Brown TH, Claiborne BJ (1993) Improved dendritic diameter measurements yield higher input resistances in modeled dentate granule neurons. *Society of Neuroscience Abstracts* 19: 799.
- Ohgane K, Ei S, Kazutoshi K, Ohtsuki T (2004) Emergence of adaptability to time delay in bipedal locomotion. *Biological Cybernetics* 90: 125–132.
- Olesen C, Picard M, Winther AM, Gyryup C, Morth JP, Oxvig C, Møller JV, Nissen P (2007) The structural basis of calcium transport by the calcium pump. *Nature* 450: 1036–1042.
- Olsen OH, Calabrese RL (1996) Activation of intrinsic and synaptic currents in leech heart interneurons by realistic waveforms. *Journal of Neuroscience* 16: 4958–4970.
- Olypher AV, Calabrese RL (2007) Using constraints on neuronal activity to reveal compensatory changes in neuronal parameters. *Journal of Neurophysiology* 98: 3749–3758.
- Olypher A, Cymbalyuk G, Calabrese RL (2006) Hybrid systems analysis of the control of burst duration by low-voltage-activated calcium current in leech heart interneurons. *Journal of Neurophysiology* 98: 3749–3758.
- O'Reilly RC, Munakata Y (2000) *Computational Explorations in Cognitive Neuroscience*. Cambridge, MA: MIT press.
- Paoletti P, Neyton J (2007) NMDA receptor subunits: Function and pharmacology. *Current Opinion in Pharmacology* 7: 39–47.
- Park KS, Mohapatra DP, Misonou H, Trimmer JS (2006) Graded regulation of the Kv2.1 potassium channel by variable phosphorylation. *Science* 313: 976–979.
- Pellionisz A, Llinás RR (1977) A computer model of cerebellar Purkinje cells. *Neuroscience* 2: 37–48.
- Percha B, Dzakpasu R, Zochowski M, Parent J (2005) Transition from local to global phase synchrony in small world neural network and its possible implications for epilepsy. *Physical Review E* 72: 031909.
- Pereda E, Quiroga RQ, Bhattacharya J (2005) Nonlinear multivariate analysis of neurophysiological signals. *Progress in Neurobiology* 77: 1–37.
- Perez JR, Ybarra N, Cantu RE, Rahimi O, Claiborne BJ (2005) Use of aged transgenic mice with GFP-expressing neurons for morphological studies. *Society for Neuroscience Abstracts* 455.6.

- Perkel DH, Mulloney B, Budelli RW (1981) Quantitative methods for predicting neuronal behavior. *Neuroscience* 4: 823–837.
- Perozo E, Bezanilla F (1990) Phosphorylation affects voltage gating of the delayed rectifier K⁺ channel by electrostatic interactions. *Neuron* 5: 685–690.
- Perttunen CD, Jones DR, Stuckman BE (1993) Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications* 79: 157–181.
- Peters A, Jones EG (1984) *Cerebral Cortex*, Vol. 1 *Cellular Components of Cerebral Cortex*. New York: Plenum.
- Pfister JP, Gerstner W (2006) Triplets of spikes in a model of spike timing-dependent plasticity. *Journal of Neuroscience* 26: 9673–9682.
- Philipson KD, Nicoll DA (2000) Sodium-calcium exchange: A molecular perspective. *Annual Review of Physiology* 62: 111–133.
- Pilwat G, Zimmermann U (1985) Determination of intracellular conductivity from electrical breakdown measurements. *Biochimica Biophysica Acta* 820: 305–314.
- Pinsky PF, Rinzel J (1994) Intrinsic and network rhythmogenesis in a reduced Traub model for CA3 neurons. *Journal of Computational Neuroscience* 1: 39–60.
- Piwkowska Z, Rudolph M, Badoval M, Destexhe A, Bal T (2005) Re-creating active states in vitro with a dynamic-clamp protocol. *Neurocomputing* 65: 55–60.
- Poggio T, Reichardt W (1976) Visual control of orientation behavior in the fly. Part II. Toward the underlying neural interactions. *Quarterly Reviews of Biophysics* 9: 377–438.
- Poirazi P, Brannon T, Mel BW (2003) Arithmetic of subthreshold synaptic summation in a model CA1 pyramidal cell. *Neuron* 37: 977–987.
- Pongracz F, Firestein S, Shepherd GM (1991) Electrotoretinal structure of olfactory sensory neurons analyzed by intracellular and whole cell patch techniques. *Journal of Neurophysiology* 65: 747–758.
- Postlethwaite M, Hennig MH, Steinert JR, Graham BP, Forsythe ID (2007) Acceleration of AMPA receptor kinetics underlies temperature-dependent changes in synaptic strength at the rat calyx of Held. *Journal of Physiology* 579: 69–84.
- Prescott SA, De Koninck Y (2003) Gain control of firing rate by shunting inhibition: Roles of synaptic noise and dendritic saturation. *Proceedings of the National Academy of Sciences of the USA* 100: 2076–2081.
- Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) *Numerical Recipes. 3rd ed. The Art of Scientific Computing*. Cambridge, UK: Cambridge University Press.
- Preyer AJ, Butera RJ (2005) Neuronal oscillators in *Aplysia californica* that demonstrate weak coupling in vitro. *Physical Review Letters* 95: 138103.
- Price KV, Stom RM, Lampinen JA (2005) *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag.
- Prinz AA, Billimoria CP, Marder E (2003) Alternative to hand-tuning conductance-based models: Construction and analysis of databases of model neurons. *Journal of Neurophysiology* 90: 3998–4015.
- Prinz AA, Bucher D, Marder E (2004a) Similar network activity from disparate circuit parameters. *Nature Neuroscience* 7: 1345–1352.
- Prinz AA, Abbott LF, Marder E (2004b) The dynamic clamp comes of age. *Trends in Neurosciences* 27: 218–224.
- Puccini GD, Sanchez-Vives MV, Compte A (2007) Integrated mechanisms of anticipation and rate-of-change computations in cortical circuits. *PLoS Computational Biology* 3: e82.
- Pugh JR, Raman IM (2005) GABA_A receptor kinetics in the cerebellar nuclei: Evidence for detection of transmitter from distant release sites. *Biophysical Journal* 88: 1740–1754.
- Purvis LK, Butera RJ (2005) Ionic current model of a hypoglossal motoneuron. *Journal of Neurophysiology* 93: 723–733.
- Pyapali GK, Sik A, Penttonen G, Buzsaki G, Turner DA (1998) Dendritic properties of hippocampal CA1 pyramidal neurons in the rat: Intracellular staining in vivo and in vitro. *Journal of Comparative Neurology* 391: 335–352.

- Qian N, Sejnowski TJ (1989) An electro-diffusion model for computing membrane potentials and ionic concentrations in branching dendrites, spines and axons. *Biological Cybernetics* 62: 1–15.
- Rabbah P, Nadim F (2005) Synaptic dynamics do not determine proper phase of activity in a central pattern generator. *Journal of Neuroscience* 25: 11269–11278.
- Radhakrishnan K, Hindmarsh AC (1993) *Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations*. Livermore, CA: Lawrence Livermore National Laboratory.
- Raghavachari S, Lisman JE (2004) Properties of quantal transmission at CA1 synapses. *Journal of Neurophysiology* 92: 2456–2467.
- Rahimi O, Ybarra N, Farina JC, Sanchez M, Claiborne BJ (2003) Dendritic trees of dentate granule neurons are larger in aged male mice than in young adults. *Society for Neuroscience Abstracts* 719.20.
- Raikov I, Preyer A, Butera RJ (2004) MRCI: A flexible real-time dynamic clamp system for electrophysiology experiments. *Journal of Neuroscience Methods* 132: 109–123.
- Rall W (1959) Branching dendritic trees and motoneuron membrane resistivity. *Experimental Neurology* 1: 491–527.
- Rall W (1962a) Theory of physiological properties of dendrites. *Annals of the New York Academy of Sciences* 96: 1071–1092.
- Rall W (1962b) Electrophysiology of a dendritic neuron model. *Biophysical Journal* 2: 145–167.
- Rall W (1964) Theoretical significance of dendritic trees for neuronal input-output relations. In RF Reiss, ed., *Neuronal Theory and Modeling*, pp. 73–97. Stanford, CA: Stanford University Press.
- Rall W (1967) Distinguishing theoretical synaptic potentials compared for different soma-dendritic distributions of synaptic input. *Journal of Neurophysiology* 30: 1138–1168.
- Rall W (1969) Time constants and electrotonic length of membrane cylinders and neurons. *Biophysical Journal* 9: 1483–1508.
- Rall W (1974) Dendritic spines, synaptic potency and neuronal plasticity. In CD Woody, KA Brown, TJ Crow, JD Knispel, eds., *Cellular Mechanisms Subserving Changes in Neuronal Activity*, pp. 13–21. Los Angeles: University of California.
- Rall W (1977) Core conductor theory and cable properties of neurons. In JM Brookhart, VB Mountcastle, eds., *Handbook of Physiology. The Nervous System. Cellular Biology of Neurons*, vol. 1, pp. 39–97. Bethesda, MD: American Physiological Society.
- Rall W (1978) Dendritic spines and synaptic potency. In R Porter, ed., *Studies in Neurophysiology*, pp. 203–209. New York: Cambridge University Press.
- Rall W, Agmon-Snir H (1998) Cable theory for dendritic neurons. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 27–92. Cambridge, MA: MIT Press.
- Rall W, Rinzel J (1973) Branch input resistance and steady attenuation for input to one branch of a dendritic neuron model. *Biophysical Journal* 13: 648–688.
- Rall W, Segev I (1985) Space-clamp problems when voltage clamping branched neurons with intracellular microelectrodes. In J Thomas, G Smith, H Lecar, SJ Redman, PW Gage, eds., *Voltage and Patch Clamping with Microelectrodes*, pp. 191–215. Bethesda MD: American Physiological Society.
- Rall W, Burke RE, Holmes WR, Jack JJB, Redman SJ, Segev I (1992) Matching neuron model complexity to experimental data. *Physiological Reviews* 72: S159–S186.
- Rao CV, Arkin AP (2003) Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *Journal of Chemical Physics* 118: 4999–5010.
- Rapp M, Yarom Y, Segev I (1992) The impact of parallel fiber background activity on the cable properties of cerebellar Purkinje cells. *Neural Computation* 4: 518–533.
- Rapp M, Segev I, Yarom Y (1994) Physiology, morphology and detailed passive models of guinea-pig cerebellar Purkinje cells. *Journal of Physiology* 474: 101–118.
- Rapp M, Yarom Y, Segev I (1996) Modeling back propagating action potential in weakly excitable dendrites of neocortical pyramidal cells. *Proceedings of the National Academy of Sciences of the USA* 93: 11985–11990.
- Rasche C, Douglas RJ (1999) Silicon synaptic conductances. *Journal of Computational Neuroscience* 7: 33–39.

- Rathinam M, Petzold LR, Cao Y, Gillespie DT (2003) Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *Journal of Chemical Physics* 119: 12784–12794.
- Ratliff F, Knight BW, Graham N (1969) On tuning and amplification by lateral inhibition. *Proceedings of the National Academy of Sciences of the USA* 62: 733–740.
- Rauch A, Camera Gl, Lüscher H-R, Senn W, Fusi S (2003) Neocortical pyramidal cells respond as integrate-and-fire neurons to in vivo-like input currents. *Journal of Neurophysiology* 190: 1598–1612.
- Rechenberg I (1973) *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Stuttgart, Germany: Frammann-Holzboog.
- Redman SJ, Walmsley B (1983) The time course of synaptic potentials evoked in cat spinal motoneurons at identified group Ia synapses. *Journal of Physiology* 343: 117–133.
- Reig R, Gallego R, Nowak LG, Sanchez-Vives MV (2006) Impact of cortical network activity on short-term synaptic depression. *Cerebral Cortex* 16: 688–695.
- Reyes AD (2003) Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro. *Nature Neuroscience* 6: 593–599.
- Reyes A, Lujan R, Rozov A, Burnashev N, Somogyi P, Sakmann B (1998) Target-cell-specific facilitation and depression in neocortical circuits. *Nature Neuroscience* 1: 279–285.
- Rhodes PA, Llinás RR (2001) Apical tuft input efficacy in layer 5 pyramidal cells from rat visual cortex. *Journal of Physiology* 536: 167–187.
- Ricciardi LM (1977) *Diffusion Processes and Related Topics on Biology*. Berlin: Springer-Verlag.
- Richardson M, Brunel N, Hakim V (2003). From subthreshold to firing-rate resonance. *Journal of Neurophysiology* 89: 2538–2554.
- Rihm LL, Claiborne BJ (1990) Dendritic growth and regression in rat dentate granule cells during late postnatal development. *Developmental Brain Research* 54: 115–124.
- Rinzel J, Ermentrout B (1998) Analysis of neural excitability and oscillations. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 251–291. Cambridge, MA: MIT Press.
- Rinzel J, Rall W (1974) Transient response in a dendritic neuron model for current injected at one branch. *Biophysical Journal* 14: 759–790.
- Roberts CB, Best JA, Suter KJ (2006) Dendritic processing of excitatory synaptic input in hypothalamic gonadotropin releasing-hormone neurons. *Endocrinology* 147: 1545–1555.
- Roberts PD (2007) Stability of complex spike timing-dependent plasticity in cerebellar learning. *Journal of Computational Neuroscience* 22: 283–296.
- Robinson DA (1992) Implications of neural networks for how we think about brain functions. *Behavioral and Brain Sciences* 15: 644–655.
- Rodriguez A, Ehlenberger D, Kelliher K (2003) Automated reconstruction of three-dimensional neuronal morphology from laser scanning microscopy images. *Methods* 30: 94–105.
- Rodriguez-Fernandez M, Mendes P, Banga JR (2006) A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems* 83: 248–265.
- Roopun AK, Middleton SJ, Cunningham MO, LeBeau FE, Bibbig A, Whittington MA, Traub RD (2006) A beta2-frequency (20–30 Hz) oscillation in nonsynaptic networks of somatosensory cortex. *Proceedings of the National Academy of Sciences of the USA* 103: 15646–15650.
- Ros E, Carrillo R, Ortigosa EM, Barbour B, Agís R (2006) Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. *Neural Computation* 18: 2959–2993.
- Rose PK, Dagum A (1988) Nonequivalent cylinder models of neurons: Interpretation of voltage transients generated by somatic current injection. *Journal of Neurophysiology* 60: 125–148.
- Rose PK, Vanner SJ (1988) Differences in somatic and dendritic specific membrane resistivity of spinal motoneurons: An electrophysiological study of neck and shoulder motoneurons in the cat. *Journal of Neurophysiology* 60: 149–166.
- Roth A, Häusser M (2001) Compartmental models of rat cerebellar Purkinje cells based on simultaneous somatic and dendritic patch-clamp recordings. *Journal of Physiology* 535: 455–472.

- Rothman JS, Cathala L, Steuber V, Silver RA (2009) Synaptic depression enables neuronal gain control. *Nature* 457: 1015–1018.
- Rotstein HG., Oppermann T, White JA, Kopell N (2006) A reduced model for medial entorhinal cortex stellate cells: Subthreshold oscillations, spiking and synchronization. *Journal of Computational Neuroscience* 21: 271–292.
- Rotter S, Diesmann M (1999). Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biological Cybernetics* 81: 381–402.
- Rudolph M, Destexhe A (2006) Analytical integrate-and-fire neuron models with conductance-based dynamics for event-driven simulation strategies. *Neural Computation* 18: 2146–2210.
- Rulkov NF (2002) Modeling of spiking-bursting neural behavior using two-dimensional map. *Physical Reviews E* 65: 041922.
- Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4: 284–294.
- Saarinen A, Linne M-L, Yli-Harja O (2008) Stochastic differential equation model for cerebellar granule cell excitability. *PLoS Computational Biology* 4: e1000004.
- Sabatini BL, Svoboda K (2000) Analysis of calcium channels in single spines using optical fluctuation analysis. *Nature* 408: 589–593.
- Sabatini BL, Maravall M, Svoboda K (2001) Ca^{2+} signaling in dendritic spines. *Current Opinion in Neurobiology* 11: 349–356.
- Sabatini BL, Oertner TG, Svoboda K (2002) The life cycle of Ca^{2+} ions in dendritic spines. *Neuron* 33: 439–452.
- Sakmann B, Neher E (eds.) (1995) *Single-Channel Recording* 2nd ed. New York: Plenum.
- Sala F, Hernandez-Cruz A (1990) Calcium diffusion modeling in a spherical neuron: Relevance of buffering properties. *Biophysical Journal* 57: 313–324.
- Salinas E, Sejnowski TJ (2000) Impact of correlated synaptic input on output firing rate and variability in simple neuronal models. *Journal of Neuroscience* 20: 6193–6209.
- Samant A, Ogunnaike BA, Vlachos DG (2007) A hybrid multiscale Monte Carlo algorithm (HyMSMC) to cope with disparity in time scales and species populations in intracellular networks. *BMC Bioinformatics* 8: 175.
- Santamaría F, Wils S, De Schutter E, Augustine GJ (2006) Anomalous diffusion in Purkinje cell dendrites caused by spines. *Neuron* 52: 635–648.
- Schaefer AT, Helmstaedter M, Sakmann B, Korngreen A (2003a) Correction of conductance measurements in non-space-clamped structures: 1. Voltage-gated K^+ channels. *Biophysical Journal* 84: 3508–3528.
- Schaefer AT, Larkum ME, Sakmann B, Roth A (2003b) Coincidence detection in pyramidal neurons is tuned by their branching pattern. *Journal of Neurophysiology* 89: 3143–3154.
- Schanne OF (1969) Measurement of cytoplasmic resistivity by means of the glass microelectrode. In M Lavallee, OF Schanne, NS Hebert, eds., *Glass Microelectrodes*, pp. 299–321. New York: Wiley.
- Schanne OF, De Ceretti ERP (1971) Measurement of input impedance and cytoplasmic resistivity with a single microelectrode. *Canadian Journal of Physiology and Pharmacology* 49: 713–716.
- Schiller J, Major G, Koester HJ, Schiller Y (2000). NMDA spikes in basal dendrites of cortical pyramidal neurons. *Nature* 404: 285–289.
- Schmidt H, Stieff KM, Racay P, Schwaller B, Eilers J (2003) Mutational analysis of dendritic Ca^{2+} kinetics in rodent Purkinje cells: Role of parvalbumin and calbindin D28k. *Journal of Physiology* 551: 13–32.
- Schmidt H, Schwaller B, Eilers J (2005) Calbindin D28k targets myo-inositol monophosphatase in spines and dendrites of cerebellar Purkinje neurons. *Proceedings of the National Academy of Sciences of the USA* 102: 5850–5855.
- Schmidt J, Calabrese RL (1992) Evidence that acetylcholine is an inhibitory transmitter of heart interneurons in the leech. *Journal of Experimental Biology* 171: 329–347.
- Schmitt S, Evers JF, Duch C, Scholz M, Obermayer K. (2004) New methods for the computer-assisted 3-D reconstruction of neurons from confocal image stacks. *Neuroimage* 23: 1283–1298.

- Schulz DJ, Goaillard JM, Marder E (2006) Variable channel expression in identified single and electrically coupled neurons in different animals. *Nature Neuroscience* 9: 356–362.
- Schwefel HP (1975) Evolutionsstrategie und numerische optimierung. PhD thesis, Technical University of Berlin.
- Scorcioni R, Ascoli GA (2005) Algorithmic reconstruction of complete axonal arborizations in rat hippocampal neurons. *Neurocomputing* 65–66: 15–22.
- Scorcioni R, Lazarewicz MT, Ascoli GA (2004) Hippocampal pyramidal cells: Differences between anatomical classes and reconstructing laboratories. *Journal of Comparative Neurology* 473: 177–193.
- Segev I, Burke RE (1998) Compartmental models of complex neurons. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 93–136. Cambridge, MA: MIT Press.
- Segev I, Parnas I (1983) Synaptic integration mechanisms: A theoretical and experimental investigation of temporal postsynaptic interaction between excitatory and inhibitory inputs. *Biophysical Journal* 41: 41–50.
- Segev I, Fleshman JW, Miller JP, Bunow B (1985) Modeling the electrical behavior of anatomically complex neurons using a network analysis program: Passive membrane. *Biological Cybernetics* 53: 27–40.
- Segev I, Rinzel J, Shepherd GM (eds.) (1994) *The theoretical Foundations of Dendritic Function: The Collected Papers of Wilfrid Rall*. Cambridge, MA: MIT Press.
- Sejnowski TJ (1977) Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology* 4: 303–321.
- Sekerli M, Butera RJ (2005) Oscillations in a simple neuromechanical system: Underlying mechanisms. *Journal of Computational Neuroscience* 19: 181–197.
- Sentft SL (1997) A statistical framework to present developmental neuroanatomy. In J Donahoe, ed., *Bio-behavioural Foundations*. Amsterdam: Elsevier.
- Shadlen MN, Newsome WT (1994) Noise, neural codes and cortical organization. *Current Opinion in Neurobiology* 4: 569–579.
- Shahrezaei V, Delaney KR (2004) Consequences of molecular-level Ca^{2+} channel and synaptic vesicle colocalization for the Ca^{2+} microdomain and neurotransmitter exocytosis: A Monte Carlo study. *Biophysical Journal* 87: 2352–2364.
- Shamma S (1998) Spatial and temporal processing in central auditory networks. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 411–460. Cambridge, MA: MIT Press.
- Shanno DF (1970) Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation* 24: 647–657.
- Shannon TR, Ginsburg KS, Bers DM (2000) Reverse mode of the sarcoplasmic reticulum calcium pump and load-dependent cytosolic calcium decline in voltage-clamped cardiac ventricular myocytes. *Biophysical Journal* 78: 322–333.
- Sharp AA, O'Neil MB, Abbott LF, Marder E (1993) Dynamic clamp: Computer-generated conductances in real neurons. *Journal of Neurophysiology* 69: 992–995.
- Sheasby BW, Fohlmeister JF (1999) Impulse encoding across the dendritic morphologies of retinal ganglion cells. *Journal of Neurophysiology* 81: 1685–1698.
- Shemer I, Brinne B, Tegnér J, Grillner S (2008) Electrotomonic signals along intracellular membranes may interconnect dendritic spines and nucleus. *PLoS Computational Biology* 4: e1000036.
- Shen GY, Wei RC, Midtgård J, Shepherd GM, Hines ML (1999) Computational analysis of action potential initiation in mitral cell soma and dendrites based on dual patch recordings. *Journal of Neurophysiology* 82: 3006–3020.
- Shepherd GM (1996) The dendritic spine: A multifunctional integrative unit. *Journal of Neurophysiology* 75: 2197–2210.
- Shepherd GM, Harris KM (1998) Three-dimensional structure and composition of CA3→CA1 axons in rat hippocampal slices: Implications for presynaptic connectivity and compartmentalization. *Journal of Neuroscience* 18: 8300–8310.
- Sherman A, Keizer J, Rinzel J (1990) Domain model for Ca^{2+} -inactivation of Ca^{2+} -channels at low channel density. *Biophysical Journal* 58: 985–995.

- Sherman SM, Guillery RW (2006) *Exploring the Thalamus and Its Role in Cortical Function*, 2nd ed. Cambridge, MA: MIT Press.
- Sholl DA (1953) Dendritic organization in the neurons of the visual and motor cortices of the cat. *Journal of Anatomy* 87: 387–406.
- Shouval HZ, Bear MF, Cooper LN (2002) A unified model of NMDA receptor-dependent bidirectional synaptic plasticity. *Proceedings of the National Academy of Sciences of the USA* 99: 10831–10836.
- Shu Y, Hasenstaub A, Duque A, Yu Y, McCormick DA (2006) Modulation of intracortical synaptic potentials by presynaptic somatic membrane potential. *Nature* 441: 761–765.
- Silberberg G, Grillner S, LeBeau FEN, Maex R, Markram, H (2005) Synaptic pathways in neural microcircuits. *Trends in Neurosciences* 28: 541–551.
- Sjöström PJ, Rancz EA, Roth A, Häusser M (2008) Dendritic excitability and synaptic plasticity. *Physiological Reviews* 88: 769–840.
- Smith GD, Cox CL, Sherman SM, Rinzel J (2000) Fourier analysis of sinusoidally driven thalamocortical relay neurons and a minimal integrate-and-fire-or-burst model. *Journal of Neurophysiology* 83: 588–610.
- Sneyd J, Falcke M (2005) Models of the inositol trisphosphate receptor. *Progress in Biophysics and Molecular Biology* 89: 207–245.
- Sobczyk A, Svoboda K (2007) Activity-dependent plasticity of the NMDA-receptor fractional Ca^{2+} current. *Neuron* 53: 17–24.
- Sohal VS, Huguenard JR (2005) Inhibitory coupling specifically generates emergent gamma oscillations in diverse cell types. *Proceedings of the National Academy of Sciences of the USA* 102: 18638–18643.
- Solinas S, Forti L, Cesana E, Mapelli J, De Schutter E, D'Angelo E (2007) Computational reconstruction of pacemaking and intrinsic electoresponsiveness in cerebellar Golgi cells. *Frontiers in Cellular Neuroscience* 1: 2.
- Solinas S, Maex R, De Schutter E (2006) Dendritic amplification of inhibitory postsynaptic potentials in a model Purkinje cell. *European Journal of Neuroscience* 23: 1207–1218.
- Soloveichik D (2009) Robust stochastic chemical reaction networks and bounded tau-leaping. *Journal of Computational Biology* 16: 501–522.
- Song S, Sjöstrom PJ, Reigl M, Nelson S, Chklovskii DB (2005) Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology* 3: e68. Erratum in *PLoS Biology* 3: e350.
- Sorensen M, DeWeerth S, Cymbalyuk G, Calabrese RL (2004) Using a hybrid neural system to reveal regulation of neuronal network activity by an intrinsic current. *Journal of Neuroscience* 24: 5427–5438.
- Sosinsky SE, Deerink TJ, Greco R, Buitenhuis CH, Bartol TM, Ellisman MH (2005) Development of a model for microphysiological simulations. *Neuroinformatics* 5: 133–162.
- Soto GE, Young SJ, Martone ME, Deerink TJ, Lamont S, Carragher BO, Hama K, Ellisman MH (1994) Serial section electron tomography: A method for three-dimensional reconstruction of large structures. *Neuroimage* 1: 230–243.
- Spain WJ, Schwindt PC, Crill WE (1987) Anomalous rectification in neurons from cat sensorimotor cortex in vitro. *Journal of Neurophysiology* 57: 1555–1576.
- Sporns O (2006) Small-world connectivity, motif composition, and complexity of fractal neuronal connections. *Biosystems* 85: 55–64.
- Spruston N, Johnston D (1992) Perforated patch-clamp analysis of the passive membrane properties of three classes of hippocampal neurons. *Journal of Neurophysiology* 97: 508–529.
- Spruston N, Jaffe DB, Williams SH, Johnston D (1993) Voltage- and space-clamp errors associated with the measurement of electrotonically remote synaptic events. *Journal of Neurophysiology* 70: 781–802.
- Stafstrom CE, Schwindt PC, Crill WE (1984) Cable properties of layer V neurons from cat sensorimotor cortex in vitro. *Journal of Neurophysiology* 52: 278–289.
- Staiger JF, Flagmeyer I, Schubert D, Ziller K, Köller R, Luhmann HJ (2004) Functional diversity of layer IV spiny neurons in rat somatosensory cortex: Quantitative morphology of electrophysiologically characterized and biocytin-labeled cells. *Cerebral Cortex* 14: 690–701.

- Staley KJ, Otis TS, Mody I (1992) Membrane properties of dentate gyrus granule cells: Comparison of sharp microelectrode and whole-cell recordings. *Journal of Neurophysiology* 67: 1346–1358.
- Staudacher, E (1998) Distribution and morphology of descending brain neurons in the cricket *Gryllus bimaculatus*. *Cell and Tissue Research* 294: 187–202.
- Stein R (1965) A theoretical analysis of neuronal variability. *Biophysical Journal* 5: 173–194.
- Steinmetz PN, Manwani A, Koch C, London M, Segev I (2000) Subthreshold voltage noise due to channel fluctuations in active neuronal membranes. *Journal of Computational Neuroscience* 9: 133–148.
- Stemmer PM, Klee CB (1994) Dual calcium-ion regulation of calcineurin by calmodulin and calcineurin-B. *Biochemistry* 33: 6859–6866.
- Steriade M (2003) *Neuronal Substrates of Sleep and Epilepsy*. Cambridge, UK: Cambridge University Press.
- Steriade M, Amzica F, Contreras D (1996) Synchronization of fast (30–40 Hz) spontaneous cortical rhythms during brain activation. *Journal of Neuroscience* 16: 392–417.
- Steuber V, Mittmann W, Hoebeek FE, Silver RA, De Zeeuw CI, Häusser M, De Schutter E (2007) Cerebellar LTD and pattern recognition by Purkinje cells. *Neuron* 54: 121–136.
- Steuber V, Schultheiss N, Silver RA, De Schutter E, Jaeger D (2009) The control of rebound spiking by voltage-gated currents and synaptic input in a detailed compartmental model of deep cerebellar nucleus neurons. (submitted).
- Stevens CF (1978) Interactions between intrinsic membrane protein and electric field. *Biophysical Journal* 22: 295–306.
- Stewart WW (1978) Functional connections between cells as revealed by dye-coupling with a highly fluorescent naphthalimide tracer. *Cell* 3: 741–759.
- Stewart I (2004) Networking opportunity. *Nature* 427: 601–604.
- Stiles JR, Bartol TM (2001) Monte Carlo methods for simulating realistic synaptic microphysiology using MCCell. In E De Schutter, ed., *Computational Neuroscience: Realistic Modeling for Experimentalists*, pp. 87–128. Boca Raton, FL: CRC Press.
- Stiles JR, Helden DV, Bartol TM, Salpeter EE, Salpeter MM (1996) Miniature endplate current rise times less than 100 microseconds from improved dual recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle. *Proceedings of the National Academy of Sciences of the USA* 93: 5747–5752.
- Stiles JR, Bartol TM, Salpeter MM, Salpeter EE, Sejnowski TJ (2001) Synaptic variability: New insights from reconstructions and Monte Carlo simulations with MCCell. In WM Cowan, TC Sudhof, CF Stevens, eds., *Synapses*, pp. 681–732. Baltimore, MD: Johns Hopkins University Press.
- Storn R, Price K (1997) Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11: 341–359.
- Strassberg A, DeFelice L (1993) Limitations of the Hodgkin-Huxley formalism: Effect of single-channel kinetics on transmembrane voltage dynamics. *Neural Computation* 5: 843–855.
- Stratford K, Mason A, Larkman A, Major G, Jack JJB (1989) The modeling of pyramidal neurons in the visual system. In R Durbin, C Miall, G Mitchison, eds., *The Computing Neuron*, pp. 296–321, Reading MA: Addison-Wesley.
- Strehler EE, Treiman M (2004) Calcium pumps of plasma membrane and cell interior. *Current Molecular Medicine* 4: 323–335.
- Strogatz S (2001) *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. New York: Perseus Books Group.
- Stuart G, Spruston N (1998) Determinants of voltage attenuation in neocortical pyramidal neuron dendrites. *Journal of Neuroscience* 18: 3501–3510.
- Sultan F, Bower JM (1998) Quantitative Golgi study of the rat cerebellar molecular layer interneurons using principal component analysis. *Journal of Comparative Neurology* 393: 353–373.
- Sun XJ, Tolbert LP, Hildebrand JG, Meinertzhagen IA (1998) A rapid method for combined laser scanning confocal microscopic and electron microscopic visualization of biocytin or neurobiotin-labeled neurons. *Journal of Histochemistry and Cytochemistry* 46: 263–274.

- Sun XP, Yazejian B, Grinnell AD (2004) Electrophysiological properties of BK channels in *Xenopus* motor nerve terminals. *Journal of Physiology* 557: 207–228.
- Sutton GP, Mangan EV, Neustadter DM, Beer RD, Crago PE, Chiel HJ (2004a) Neural control exploits changing mechanical advantage and context dependence to generate different feeding responses in *Aplysia*. *Biological Cybernetics* 91: 333–345.
- Sutton GP, Macknin JB, Gartman SS, Sunny GP, Beer RD, Crago PE, Neustadter DM, Chiel HJ (2004b) Passive hinge forces in the feeding apparatus of *Aplysia* aid retraction during biting but not during swallowing. *Journal of Comparative Physiology A* 190: 501–514.
- Svirskis G, Gutman A, Hounsgaard J (2001) Electrotonic structure of motoneurons in the spinal cord of the turtle: Inferences for the mechanisms of bistability. *Journal of Neurophysiology* 85: 391–398.
- Swensen AM, Bean BP (2005) Robustness of burst firing in dissociated Purkinje neurons with acute or long-term reductions in sodium conductance. *Journal of Neuroscience* 25: 3509–3520.
- Szilagyi T, De Schutter E (2004) Effects of variability in anatomical reconstruction techniques on models of synaptic integration by dendrites: A comparison of three Internet archives. *European Journal of Neuroscience* 19: 1257–1266.
- Tadross MR, Dick IE, Yue DT (2008) Mechanism of local and global Ca^{2+} sensing by calmodulin in complex with a Ca^{2+} channel. *Cell* 133: 1228–1240.
- Takahashi A, Camacho P, Lechleiter JD, Herman B (1999) Measurement of intracellular calcium. *Physiological Reviews* 79: 1089–1125.
- Takahashi T (2005) Postsynaptic receptor mechanisms underlying developmental speeding of synaptic transmission. *Neuroscience Research* 53: 229–240.
- Takahashi T, Forsythe ID, Tsujimoto T, Barnes-Davies M, Onodera K (1996) Presynaptic calcium current modulation by a metabotropic glutamate receptor. *Science* 274: 594–597.
- Takeuchi A, Tatsumi S, Sarai N, Terashima K, Matsuoka S, Noma A (2006) Ionic mechanisms of cardiac cell swelling induced by blocking Na^+/K^+ pump as revealed by experiments and simulation. *Journal of General Physiology* 128: 495–507.
- Tao L, Cai D, McLaughlin DW, Shelley MJ, Shapley R (2006) Orientation selectivity in visual cortex by fluctuation-controlled criticality. *Proceedings of the National Academy of Sciences of the USA* 103: 12911–12916.
- Taylor AL, Hickey TJ, Prinz AA, Marder E (2006) Structure and visualization of high-dimensional conductance spaces. *Journal of Neurophysiology* 96: 891–905.
- Taylor CW, Laude AJ (2002) IP₃ receptors and their regulation by calmodulin and cytosolic Ca^{2+} . *Cell Calcium* 32: 321–334.
- Teeters JL, Harris KD, Millman KJ, Olshausen BA, Sommer FT (2008) Data sharing for computational neuroscience. *Neuroinformatics* 6: 47–55.
- Tegnér J, Grillner S (1999) Interactive effects of the GABA_Bergic modulation of calcium channels and calcium-dependent potassium channels in lamprey. *Journal of Neurophysiology* 81: 1318–1329.
- Thomson AM, Lamy C (2007) Functional maps of neocortical local circuitry. *Frontiers in Neuroscience* 1: 19–42.
- Thomson AM (1997) Activity-dependent properties of synaptic transmission at two classes of connections made by rat neocortical pyramidal axons in vitro. *Journal of Physiology* 502: 131–147.
- Thomson AM, Deuchars J, West DC (1993) Large, deep layer pyramid-pyramid single axon EPSPS in slices of rat motor cortex display paired pulse and frequency-dependent depression, mediated presynaptically, and self-facilitation, mediated postsynaptically. *Journal of Neurophysiology* 70: 2354–2369.
- Thurbon D, Luscher H-R, Hofstetter T, Redman SJ (1998) Passive electrical properties of ventral horn neurons in rat spinal cord slices. *Journal of Neurophysiology* 79: 2485–2502.
- Tian TH, Burrage K (2004) Binomial leap methods for simulating stochastic chemical kinetics. *Journal of Chemical Physics* 121: 10356–10364.
- Tobin AE, Calabrese RL (2005) Myomodulin increases Ih and inhibits the NA/K pump to modulate bursting in leech heart interneurons. *Journal of Neurophysiology* 94: 3938–3950.

- Tomita S, Adesnik H, Sekiguchi M, Zhang W, Wada K, Howe JR, Nicoll RA, Bredt DS (2005) Stargazin modulates AMPA receptor gating and trafficking by distinct domains. *Nature* 435: 1052–1058.
- Torben-Nielsen B, Vanderlooy S, Postma EO (2008) Non-parametric algorithmic generation of neuronal morphologies. *Neuroinformatics* 6: 257–277.
- Touboul J, Brette R (2008) Dynamics and bifurcations of the adaptive exponential integrate-and-fire model. *Biological Cybernetics* 99: 319–334.
- Traub RD (1979) Neocortical pyramidal cells: A model with dendritic calcium conductance reproduces repetitive firing and epileptic behavior. *Brain Research* 173: 243–257.
- Traub RD, Llinás R (1977) The spatial distribution of ionic conductances in normal and axotomized motoneurons. *Neuroscience* 2: 829–850.
- Traub RD, Wong RKS, Miles R, Michelson H (1991) A model of a CA3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances. *Journal of Neurophysiology* 66: 635–650.
- Traub RD, Jefferys JG, Whittington MA (1999a) *Fast Oscillations in Cortical Circuits*. Cambridge, MA: MIT Press.
- Traub RD, Schmitz D, Jefferys JG, Draguhn A (1999b) High-frequency population oscillations are predicted to occur in hippocampal pyramidal neuronal networks interconnected by axoaxonal gap junctions. *Neuroscience* 92: 407–426.
- Traub RD, Contreras D, Cunningham MO, Murray H, LeBeau FE, Roopun A, Bibbig A, Wilent WB, Higley MJ, Whittington MA (2005) Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. *Journal of Neurophysiology* 93: 2194–2232.
- Trevelyan AJ, Jack J (2002) Detailed passive cable models of layer 2/3 pyramidal cells in rat visual cortex at different temperatures. *Journal of Physiology* 539: 623–636.
- Treves A (1993) Mean-field analysis of neuronal spike dynamics. *Network* 4: 259–284.
- Troyer TW, Miller KD (1997) Physiological gain leads to high ISI variability in a simple model of a cortical regular spiking cell. *Neural Computation* 9: 971–983.
- Trussell LO (1999) Synaptic mechanisms for coding timing in auditory neurons. *Annual Review of Physiology* 61: 477–496.
- Tsien RW and Noble D (1969) A transition state theory approach to the kinetics of conductances in excitable membranes. *Journal of Membrane Biology* 1: 248–273.
- Tsodyks MV, Markram H (1997) The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences of the USA* 94: 719–723.
- Tsodyks M, Sejnowski TJ (1995) Rapid switching in balanced cortical network models. *Network* 6: 1–14.
- Tsodyks MV, Pawelzik K, Markram H (1998) Neural networks with dynamic synapses. *Neural Computation* 10: 821–835.
- Tsodyks M, Uziel A, Markram H (2000) Synchrony generation in recurrent networks with frequency-dependent synapses. *Journal of Neuroscience* 20: RC50.
- Tuckwell HC (1988) *Introduction to Theoretical Neurobiology*. Cambridge, UK: Cambridge University Press.
- Tuckwell HC (1995) *Elementary Applications of Probability Theory*. London : Chapman and Hall.
- Turner DA, Schwartzkroin PA (1980) Steady-state electrotonic analysis of intracellularly stained hippocampal neurons. *Journal of Neurophysiology* 44: 184–199.
- Tyler MN, Shepherd D, Williams DW (2000) Methods for imaging labeled neurons together with neuropil features in *Drosophila*. *Journal of Histochemistry and Cytochemistry* 48: 1575–1581.
- Van Geit W, Achard P, De Schutter E (2007) Neurofitter: A parameter tuning package for a wide range of electrophysiological neuron models. *Frontiers in Neuroinformatics* 1: 1.
- van Kampen NG (1981) *Stochastic Processes in Physics and Chemistry*. Amsterdam: North-Holland.
- Van Ooyen A, Van Pelt J (2002) Competition in neuronal morphogenesis and the development of nerve connections. In GA Ascoli, ed., *Computational Neuroanatomy: Principles and Methods*, pp. 219–244. Totowa, NJ: Humana.

- Van Ooyen A, Duijnhouwer J, Remme MWH, Van Pelt J (2002) The effect of dendritic topology on firing patterns in model neurons. *Network: Computation in Neural Systems* 13: 311–325.
- Van Pelt J, Schierwagen AK (1994) Electrotonic properties of passive dendritic trees—effect of dendritic topology. *Progress in Brain Research* 102: 127–149.
- Van Pelt J, Schierwagen AK (1995) Dendritic electrotonic extent and branching pattern topology. In JM Bower, ed., *The Neurobiology of Computation*, pp. 153–158. Boston: Kluwer.
- van Rossum MCW, Smith RG (1998) Noise removal at the rod synapse of mammalian retina. *Visual Neuroscience* 15: 809–821.
- van Rossum MC, Turrigiano GG, Nelson SB (2002) Fast propagation of firing rates through layered networks of noisy neurons. *Journal of Neuroscience* 22: 1956–1966.
- van Rossum MCW, O'Brien BJ, Smith RG (2003) The effects of noise on the timing precision of retinal ganglion cells. *Journal of Neurophysiology* 89: 2406–2419.
- van Rossum MCW, van der Meer MAA, Xiao D, Oram MW (2008) Adaptive integration in the visual cortex by depressing recurrent cortical circuits. *Neural Computation* 20: 1847–1872.
- van Vreeswijk C, Abbott LF, Ermentrout GB (1994) When inhibition not excitation synchronizes neural firing. *Journal of Computational Neuroscience* 1: 313–321.
- van Vreeswijk C, Sompolinsky H (1996) Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* 274: 1724–1726.
- van Vreeswijk C, Hansel D (2001) Patterns of synchrony in neural networks with spike adaptation. *Neural Computation* 13: 959–992.
- Vandenberg CA, Bezanilla F (1991) A model of sodium channel gating based on single channel, macroscopic ionic, and gating currents in the squid giant axon. *Biophysical Journal* 60: 1511–1533.
- Vanier MC, Bower JM (1999) A comparative survey of automated parameter-search methods for compartmental neural models. *Journal of Computational Neuroscience* 7: 149–171.
- Varela JA, Sen K, Gibson J, Fost J, Abbott LF, Nelson SB (1997) A quantitative description of short-term plasticity at excitatory synapses in layer 2/3 of rat primary visual cortex. *Journal of Neuroscience* 17: 7926–7940.
- Vargas-Caballero M, Robinson HPC (2003) A slow fraction of Mg^{2+} unblock of NMDA receptors limits their contribution to spike generation in cortical pyramidal neurons. *Journal of Neurophysiology* 89: 2778–2783.
- Vargas-Caballero M, Robinson HPC (2004) Fast and slow voltage-dependent dynamics of magnesium block in the NMDA receptor: The asymmetric trapping block model. *Journal of Neuroscience* 24: 6171–6180.
- Varjú D (1962) Vergleich zweier modelle für die laterale inhibition. *Kybernetik* 1: 200–208.
- Vasudeva K, Bhalla US (2004) Adaptive stochastic-deterministic chemical kinetic simulations. *Bioinformatics* 20: 78–84.
- Vervaeke K, Hu H, Graham LJ, Storm JF (2006) Contrasting effects of the persistent Na^+ current on neuronal excitability and spike timing. *Neuron* 49: 257–270.
- Vesterstrom J, Thomsen R (2004) A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Proceedings of 2004 Congress on Evolutionary Computation*, pp. 332–339. Piscataway, NJ: IEEE Press.
- Vetter P, Roth A, Häusser M (2001) Propagation of action potential in dendrites depends on dendritic morphology. *Journal of Neurophysiology* 85: 926–937.
- Vida I, Bartos M, Jonas P (2006) Shunting inhibition improves robustness of gamma oscillations in hippocampal interneuron networks by homogenizing firing rates. *Neuron* 49: 107–117.
- Vilar JM, Guet CC, Leibler S (2003) Modeling network dynamics: The lac operon, a case study. *Journal of Cellular Biology* 161: 471–476.
- Vogel R, Weingart R (1998) Mathematical model of vertebrate gap junctions derived from electrical measurements on homotypic and heterotypic channels. *Journal of Physiology* 510: 177–189.

- Vogels TP, Abbott LF (2005) Signal propagation and logic gating in networks of integrate-and-fire neurons. *Journal of Neuroscience* 25: 10786–10786.
- Vogels TP, Rajan K, Abbott LF (2005) Neural network dynamics. *Annual Review of Neuroscience* 28: 357–376.
- Vos BP, Maex R, Volny-Luraghi A, De Schutter E (1999) Parallel fibers synchronize spontaneous activity in cerebellar Golgi cells. *Journal of Neuroscience* 19: RC6.
- Wagner J, Keizer J (1994) Effects of rapid buffers on Ca^{2+} diffusion and Ca^{2+} oscillations. *Biophysical Journal* 67: 447–456.
- Walker HC, Lawrence JJ, McBain CJ (2002) Activation of kinetically distinct synaptic conductances on inhibitory interneurons by electrotonically overlapping afferents. *Neuron* 35: 161–171.
- Wang S-S, Denk W, Häusser M (2000) Coincidence detection in single spines mediated by calcium release. *Nature Neuroscience* 3: 1266–1273.
- Wang X-J (1993) Ionic basis for intrinsic oscillations. *NeuroReport* 5: 221–224.
- Wang X-J, Buzsaki G (1996) Gamma oscillation by synaptic inhibition in a hippocampal interneuronal network model. *Journal of Neuroscience* 16: 6402–6413.
- Wang Y, Gupta A, Toledo-Rodriguez M, Wu CZ, Markham H (2002) Anatomical, physiological, molecular and circuit properties of nest basket cells in the developing somatosensory cortex. *Cerebral Cortex* 12: 395–410.
- Wang Y, Toledo-Rodriguez M, Gupta A, Wu C, Silberberg G, Luo J, Markram H (2004) Anatomical, physiological and molecular properties of Martinotti cells in the somatosensory cortex of the juvenile rat. *Journal of Physiology* 561: 65–90.
- Wang Y, Markram H, Goodman PH, Berger TK, Ma J, Goldman-Rakic PS (2006) Heterogeneity in the pyramidal network of the medial prefrontal cortex. *Nature Neuroscience* 9: 534–542.
- Watanabe S, Hoffman DA, Migliore M, Johnston D (2002) Dendritic K^+ channels contribute to spike-timing dependent long-term potentiation in hippocampal pyramidal neurons. *Proceedings of the National Academy of Sciences of the USA* 99: 8366–8371.
- Weaver CM, Wearne SL (2006) The role of action potential shape and parameter constraints in optimization of compartment models. *Neurocomputing* 69: 1053–1057.
- Weaver CM, Wearne SL (2008) Neuronal firing sensitivity to morphologic and active membrane parameters. *PLoS Computational Biology* 4: e1.
- Weizenbaum J (1976) *Computer Power and Human Reason*. San Francisco: W.H. Freeman.
- White EL, Rock MP (1980) Three-dimensional aspects and synaptic relationships of a Golgi-impregnated spiny stellate cell reconstructed from serial thin sections. *Journal of Neurocytology* 9: 615–636.
- Wikström M, Morgan E, Verkhovsky MI (1997) Proton and electrical charge translocation by cytochrome c oxidase. *Biochimica Biophysica Acta* 1318: 299–306.
- Williams SR, Mitchell SJ (2008) Direct measurement of somatic voltage clamp errors in central neurons. *Nature Neuroscience* 11: 790–798.
- Willms AR, Baro DJ, Harris-Warrick RM, Guckenheimer J (1999) An improved parameter estimation method for Hodgkin-Huxley models. *Journal of Computational Neuroscience* 6: 145–168.
- Wilson HR, Blake R, Lee SH (2001) Dynamics of travelling waves in visual perception. *Nature* 412: 907–910.
- Wilson M, Bower JM (1989) The simulation of large-scale neural networks. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Synapses to Networks*, pp. 291–333. Cambridge, MA: MIT Press.
- Wilson M, Bower JM (1992) Cortical oscillations and temporal interactions in a computer simulation of piriform cortex. *Journal of Neurophysiology* 67: 981–995.
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1: 67–82.
- Wörgötter F, Koch C (1991) A detailed model of the primary visual pathway in the cat: Comparison of afferent excitatory and intracortical inhibitory connection schemes for orientation selectivity. *Journal of Neuroscience* 11: 1959–1979.

- Xing J, Gerstein GL (1996) Networks with lateral connectivity. I. Dynamic properties mediated by the balance of intrinsic excitation and inhibition. *Journal of Neurophysiology* 75: 184–199.
- Yamada WM, Koch C, Adams PR (1998) Multiple channels and calcium dynamics. In C Koch, I Segev, eds., *Methods in Neuronal Modeling: From Ions to Networks*, pp. 137–170. Cambridge, MA: MIT Press.
- Yamashita M, Sugioka M, Ogawa Y (2006) Voltage- and Ca^{2+} -activated potassium channels in Ca^{2+} store control Ca^{2+} release. *FEBS Journal* 273: 3585–3597.
- Yamazaki T, Tanaka S (2007) A spiking network model for passage-of-time representation in the cerebellum. *European Journal of Neuroscience* 26: 2279–2292.
- Yano K, Petersen OH, Tepikin AV (2004) Dual sensitivity of sarcoplasmic/endoplasmic Ca^{2+} -ATPase to cytosolic and endoplasmic reticulum Ca^{2+} as a mechanism of modulating cytosolic Ca^{2+} oscillations. *Biochemical Journal* 383: 353–360.
- Young JM, Waleszczyk WJ, Wang C, Calford MB, Dreher B, Obermayer K (2007) Cortical reorganization consistent with spike timing—but not correlation-dependent plasticity. *Nature Neuroscience* 10: 887–895.
- Zador A, Koch C (1994) Linearized models of calcium dynamics: Formal equivalence to the cable equation. *Journal of Neuroscience* 14: 4705–4715.
- Zador A, Koch C, Brown TH (1990) Biophysical model of a Hebbian synapse. *Proceedings of the National Academy of Sciences of the USA* 87: 6718–6722.
- Zador AM, Agmon-Snir H, Segev I (1995) The morphoelectrotonic transform: A graphical approach to dendritic function. *Journal of Neuroscience* 15: 1660–1682.
- Zahradník I, Gyorke S, Zahradníková A (2005) Calcium activation of ryanodine receptor channels—reconciling RyR gating models with tetrameric channel structure. *Journal of General Physiology* 126: 515–527.
- Zhang Y, Oliva R, Gisselmann N, Hatt H, Guckenheimer J, Harris-Warrick RM (2003) Overexpression of a hyperpolarization-activated cation current (I-h) channel gene modifies the firing activity of identified motor neurons in a small neural network. *Journal of Neuroscience* 23: 9059–9067.
- Zhang Z, Deschenes M (1997) Intracortical axonal projections of lamina VI cells of the primary somatosensory cortex in the rat: A single-cell labeling study. *Journal of Neuroscience* 17: 6365–6379.
- Zhou L, Chiu SY (2001) Computer model for action potential propagation through branch point in myelinated nerves. *Journal of Neurophysiology* 85: 197–210.
- Zhou Y, Carney LH, Colburn HS (2005) A model for interaural time difference sensitivity in the medial superior olive: Interaction of excitatory and inhibitory synaptic inputs, channel dynamics, and cellular morphology. *Journal of Neuroscience* 25: 3046–3058.
- Zhuge R, Fogarty KE, Tuft RA, Walsh JV (2002) Spontaneous transient outward currents arise from microdomains where BK channels are exposed to a mean Ca^{2+} concentration on the order of 10 microM during a Ca^{2+} spark. *Journal of General Physiology* 120: 15–27.
- Zou Q, Bornat Y, Tomas J, Renaud S, Destexhe A (2006) Real-time simulations of networks of Hodgkin-Huxley neurons using analog circuits. *Neurocomputing* 69: 1137–1140.
- Zucker RS, Regehr WG (2002) Short-term synaptic plasticity. *Annual Review of Physiology* 64: 355–405.

Contributors

Pablo Achard

Volen Center for Complex Systems
Brandeis University
Waltham, Massachusetts, USA
and Laboratory of Theoretical Neurobiology
University of Antwerp, Belgium

Haroon Anwar

Brain Mind Institute, EPFL
Lausanne, Switzerland
and Computational Neuroscience Unit
Okinawa Institute of Science and Technology, Japan

Michiel Berends

Laboratory of Theoretical Neurobiology
University of Antwerp, Belgium

Upinder S. Bhalla

Neurobiology, National Centre for Biological Sciences, TIFR
Bangalore, India

Nicolas Brunel

Laboratory of Neurophysics and Physiology
Université Paris Descartes, CNRS
Paris, France

Ronald L. Calabrese

Department of Biology
Emory University
Atlanta, Georgia, USA

Brenda Claiborne

College of Arts and Sciences
University of New Mexico
Albuquerque, New Mexico, USA

Hugo Cornelis

Research Imaging Center
University of Texas Health Science Center at San Antonio, Texas, USA

Erik De Schutter

Computational Neuroscience Unit
Okinawa Institute of Science and Technology, Japan
and Laboratory of Theoretical Neurobiology
University of Antwerp, Belgium

Alain Destexhe

Laboratory for Computational Neuroscience, UNIC, CNRS
Gif-sur-Yvette, France

Bard Ermentrout

Department of Mathematics
University of Pittsburgh
Pittsburgh, Pennsylvania, USA

Kristen Harris

Center for Learning and Memory
University of Texas, Austin, Texas, USA

Sean Hill

Brain Mind Institute, EPFL
Lausanne, Switzerland

John R. Huguenard

Stanford University Medical Center
Stanford, California, USA

William R. Holmes

Department of Biological Sciences
Neuroscience Program and Quantitative Biology Institute
Ohio University, Athens, Ohio, USA

Gwen Jacobs

Center for Computational Biology
Montana State University, Bozeman, Montana, USA

Gwendal LeMasson

Laboratoire de Physiopathologie des Réseaux Neuronaux Médullaires, INSERM
Université Bordeaux, France

Henry Markram

Brain Mind Institute, EPFL
Lausanne, Switzerland

Reinoud Maex

Laboratory of Theoretical Neurobiology
University of Antwerp, Belgium

Astrid A. Prinz

Department of Biology
Emory University
Atlanta, Georgia, USA

Imad Riachi

Brain Mind Institute, EPFL
Lausanne, Switzerland

John Rinzel

Center for Neural Science
New York University
New York, New York, USA

Arnd Roth

Wolfson Institute for Biomedical Research
University College London, UK

Felix Schürmann

Brain Mind Institute, EPFL
Lausanne, Switzerland

Werner Van Geit

Computational Neuroscience Unit
Okinawa Institute of Science and Technology, Japan
and Laboratory of Theoretical Neurobiology
University of Antwerp, Belgium

Mark C. W. van Rossum

Institute for Adaptive and Neural Computation
University of Edinburgh, UK

Stefan Wils

Computational Neuroscience Unit
Okinawa Institute of Science and Technology, Japan
and Laboratory of Theoretical Neurobiology
University of Antwerp, Belgium

Index

Note: boldface page numbers indicate main or defining entry for a term.

- 3/2 rule. *See* Branch
- Accuracy, 24–26, 77–79, 99, 263–264, 271
- Action potential initiation, 274
model of, **12–23**, 100, 108–111, 123–125, 127–128
phase-plane analysis, 12–17
role in transmitter release, 148, 153
- Activated complex, 117–118
- Activation. *See* Channel
- Active membrane model, **259–283**, 286–289, 326–327
- Activity ratio, 90
- Adaptation, 182, 325–327, 353
- Adaptive EIF. *See* Integrate-and-fire neuron model, adaptive exponential
- Adaptive step. *See* Time step, adaptive
- Affinity, 64
- Alexa dye, 188, 191
- ALIF. *See* Integrate-and-fire neuron model, adaptive
- Alignment, 203–204
- Alpha function. *See* Synapse model
- Alternating direction implicit method, 99
- Amira software, 203, 205, 206
- AMPA receptor, 63–64, 140, 144, **146**, 149
EPSC, 146
model, 150–151
regulation, 149
temperature sensitivity, 145
- Amplification, 62, 350
- Apical dendrite, 218–220
- Apical tuft, 218–220, 231
- Artifact, 194–195, 213, 340
edge, 339
- Artificial conductance, 306, 309–310
- Artificial synapse, 308, 311–312
- Asymmetry, 231
- Attenuation. *See* Voltage, attenuation
- Attractor, 10
global, 14–15
- Autocorrelation, 342
- Automated parameter fitting. *See* Parameter fitting
- Autonomous. *See* Differential equation, autonomous
- Average. *See* Mean value
- Axial current, 100, 235–236, 261
- Axon, 274–275, 327–328
delay, 148, 312, 321, 329–332, 347, 353
model, 18–20, 275, 327–328
hillock, 274
initial segment, 274, 276
morphology, 274–275, 328
myelinated, 274
repair of morphology, 227–230
- Balanced network. *See* Inhibition, balanced
- BCM rule. *See* Bienenstock-Cooper-Munro rule
- Beads, 189
- Bernoulli process, 87
- Bienenstock-Cooper-Munro rule, 157
- Bifurcation, **10–11**, 15–17
branch (*see* Branch)
diagram, 10–11
Hopf, **15–17**, 169, 171
saddle node, **16–17**, 169, 171
SNIC (*see* saddle node)
- Binding step, 64. *See also* Stoichiometry
- Binomial distribution, 155–156
- Biocytin, 188–191
- Biophysical mechanism, 271
- Biophysical model, 65
- Biotinylated dextran amine, 192
- Bistability, 10–11, 17, 62
- BK-type channel. *See* Potassium channel, BK-type
 Ca^{2+} -activated
- Blebbing, 191, 194–195
- Boltzmann equation, **31–32**, 111, 128
- Boltzmann-Gibbs distribution, 45

- Borg-Graham equation, 120
 Boundary, 86–88
 Boundary condition, 18, 20, 29, 82–83, 237–239, 339
 Neumann-type, 83
 volumetric, 220–223, 231
 Boundary-value problem, 21, 24, 26–27
 Branch
 3/2 rule, 239–340, 245
 axial resistance, 263
 bifurcation, 213, 215, 221–225
 continuation, 215
 cut, 195–196, 218–227, 231
 definition, 214
 impedance match, 245
 order, 213, 215–216, 219, 221
 order-based algorithm, 213
 termination, 213, 215, 221–225
 Briggs-Haldane equation, 67
 Brownian motion, 27, 85–86
 Buffer, 97–99, 102–104
 capacity, 98, 103
 concentration, 98
 mobile, 98, 103–104
 saturation, 103
 Bursting, 180–184
 Ca-ATPase pump, 95–96
 Ca^{2+} . *See* Calcium
 Cable
 equation, 28–29, 100, 236–237
 equivalent, 250–251, 265 (*see also* Cylinder)
 model, 236–237, 259, 263
 nonuniform parameters, 242, 248–250, 257
 parameter, 239, 246–251, 255–257, 261
 theory, 233–246
 Calbindin, 98, 102
 CalC software, 99, 355
 Calcium
 -activated channel (*see* Potassium channel,
 Ca^{2+} -activated)
 activation by, 260, 265
 buffer (*see* Buffer)
 channel, 148, 154, 261
 concentration, 69, 93–105, 148, 152, 154, 279
 current, 93, 148, 293, 295
 diffusion, 99, 103–104
 dye, 102, 104
 fractional current, 94, 145
 influx, 63, 93–94, 100–103, 145, 148, 158
 low-threshold current, 111–115, 127–136
 oscillation, 96–97
 pool, 101–102, 282
 pump, 103 (*see also* Ca-ATPase pump)
 release, 96–97
 removal, 95–96, 101
 spark, 101
 spike, 135–136
 store, 96–98, 104
 T-type current, 111–115, 127–136
 wave, 96
 Calibration, 204
 Calmodulin, 63–64, 98
 Calreticulin, 98
 Calsequestrin, 98
 CaMKII, 64, 158
 Canonical model, 271, 289–291, 296–297
 Capacitance, 1
 load, 274
 membrane, 234–235, 246, 250, 255–256, 260–261
 Carnoy's fixative, 191
 Catacomb software, 355
 CCH. *See* Cross-correlation histogram
 Cell Centered Database, 209
 Central pattern generator, 289–291, 310, 315–316.
 See also Leech, heartbeat timing network
 Central peak, 343
 Cerebellar cortex model, 336
 Channel 107–137. *See also* State; Thermodynamic model
 activation, 109–113, 115, 120, 125, 128–130, 133–134
 activation gate, 110, 115, 123
 calcium (*see* Calcium, channel)
 conductance, 109–110, 115, 263–264
 conformational change, 108, 117–122, 132
 data, 137, 274
 deactivation, 112–113, 121
 deinactivation, 112, 114
 density, 131, 136, 263, 265, 276
 gating, 109–112, 117–123, 149, 154, 260, 279
 inactivation, 110–116, 123–127, 133–134
 kinetic equation, 122–123
 kinetic table, 264–265, 350
 kinetics, 109–114, 117–121, 274–276 (*see also* Parameter fitting)
 leak (*see* Leak)
 permeability, 116, 145
 potassium (*see* Potassium channel)
 rate constant, 22, 109–110, 117–119, 122, 127
 single, 107–108, 121–124
 sodium (*see* Sodium channel)
 stochastic model, 22–23, 27–28, 121–123, 275
 time constant, 111, 113–115, 120–121, 128–130, 132–134
 ChannelDB database, 137
 Characteristic polynomial, 6–7
 Charge, 94, 109, 116
 equalization, 247
 Chemical master equation, 72–74
 Chi-square test, 33, 59
 CICR. *See* Calcium, release
 Closed-loop system, 313–316
 Clustering, 333
 cm ; C_m . *See* Capacitance, membrane
 CME. *See* Chemical master equation

- Cobalt chloride, 188, 190–191
Coefficient of variation, 167–168, 342
Coevolution, 55
Collision, 69, 71, 85–86
Collocation, 26–27
Communication, 348
Compartment
 asymmetric, 261, 263
 chemical, 66
 isopotential, 24, 109, 240, **259**–260, 263
 size, 263–264
 symmetric, 261, 263
Compartmental model, 18, 240, 247, **259**–**265**, 266–267, 269–282
 accuracy, 263–264
 chemical, 66
 parameter, 261–263 (*see also* Parameter fitting)
 solution method, 264–265
 surface area, 202
Computation
 chemical, 61–63
 function of, 318
Concentration, 66–68, 81–82, 116. *See also* Calcium
Conductance. *See also* Channel; Synaptic conductance
-based model, 161, **285**–**288**, 295–297
coupling, 267
input (*see* Resistance, input)
maximal, **110**, 263, 288–290, 295–296, 299–300, 304
Conducting state, 153
Conduction delay. *See* Axon, delay
Confocal image, 200–201
Conformational change. *See* Channel
Conjugate gradient method, 39, 58, 255
Connection, 288, 324–325
 delay (*see* Axon)
 density, 335
 topology, 333–334
Constant field equation. *See* Goldman-Hodgkin-Katz equation
Constraint, 56–57
Continuation, 215
Convolution, 322–323
Coordination, 294, 297–298, 314–315
 intersegmental, 316
Core conductor, 233
Cost function. *See* Fitness function
CPG. *See* Central pattern generator
CPU, 347–350
Crank-Nicholson method, 99, 264, 340
Cricket interneuron, 190–191, 208
Cross-correlation histogram, 342–344
Current, 107–116
 cancellation, 310
 fitting, 123–136
 gating current, 124
 injection, 234, 238, 240–241, 252, **259**–**261**, 286
 (*see also* Input)
 leak (*see* Leak)
membrane, **234**–**235**, 260–261, 286–288
ohmic, **116**, 144, 260–261
synapse model, 144
synaptic, 143–145
tail, **112**–**113**, 115
threshold, 176
CV. *See* Coefficient of variation
CVapp software, 206–207
CVODE software, 26
Cylinder, 261–263
 equivalent, **239**–**240**, 245–246, 257
infinite, 237–239
Database, 137, 208–210, 305
 modeling approach, 300–305
DE. *See* Direct evolution
Deactivation. *See* Channel
Decay. *See* Synaptic conductance; Voltage
Decorrelation, 153
Dehydration, 190–191, 195, 197
Deinactivation. *See* Channel
Delay, 315. *See also* Axon
Delay-differential equation, 4
Delayed rectifier. *See* Potassium channel
Dendrite, 266. *See also* Branch; Dendritic spine;
 Spike
filtering, 243
growth model, 213–214
repair of morphology, 214–227
topology, 261, 263
unraveling, 216–217, 230
Dendritic spine, 64, 69, 94, 187–189, 196–197, 202, 245–246
 compensation, **253**–**255**, 280
 density, 99–100, 253, 274, 280
 model, 99–100, 102–104
Depression. *See* Synaptic depression
Desensitization, 63. *See also* Synaptic conductance
Deterministic solution method. *See* Diffusion;
 Reaction
Deterministic search method, 42–44
Development, 145–146, 149
Dialysis, 248
Diameter, 195, 198–199, 201–203, 238–239, 243–245, 255–256
Diaminobenzidine, 190, 195
Differential equation, 1–30
 autonomous, 4–7
 homogeneous, 5
 linear, 5–8
 order, 3–4
 ordinary, 1, 2–4, 21, 24–26, 29
 partial, **20**–**21**, 24, 28–29
 stochastic, 27, 78–80
Differential evolution, **49**–**50**, 58

- Differentiation, 63
 Diffusion, **80–83**, 85–86, 99–100
 1-D, 99, 103–104
 3-D, 99–100, 104–105
 anomalous diffusion, 99–100
 apparent diffusion constant, **99**, 103–104
 approximation, 167
 calcium (*see* Calcium)
 constant **81–82**, 85–87, 98–99, 149
 deterministic solution method, 83–85
 electrodiffusion, 100
 growth model, 214
 in membrane, 154
 lateral, 154
 numerical integration, 83–85, 99–100
 stochastic solution method, 85–88
 transmitter, 145, 148–149, 155
 Dil labeling, 192–193
 Dimensional stacking, 302–304
 Dimensionality, 99
 DIRECT algorithm, **42–44**, 58
 Direction vector, 216
 Discrete-event simulation, 147–148, 348
 Discretization, 83, 87–88, 263–264
 Disinhibition, 351
 Dissociation constant. *See* Reaction
 Dominance, 53–54
 Dose-response curve, 90
 Downhill simplex method, 39–40
 DRéAMM software, 206, **358**
 Driving force, 115, **144**, 148, 241, 244
 Dye leakage, 213
 Dynamic clamp, 278, **305–312**
- E-Cell software, 356
 EA. *See* Evolutionary algorithm
 EEG, 341
 EIF. *See* Integrate-and-fire neuron model, exponential
 Eigenfunction, 247
 Eigenvalue, **6–7**, 9, 15, 247
 Eigenvector, 6
 Electric field, 100, 117–120, 273, 338, 341
 Electrical coupling. *See* Gap junction
 Electrodiffusion, 100
 Electron microscopy, 196–199, 202–203
 Electron tomography, 196, 205–206, 210
 tracing, 206–207
 Electrophysiological data, 33–34, 36, 251–253,
 257–258, 271–273, 343–344. *See also* Patch-clamp recording; Sharp electrode recording
 Electrotonic
 distance, 237
 length, **237–239**, 244–246, 250, 263–264
 parameter (*see* Cable)
 Endoplasmic reticulum, 83, 85, 95–96, 104
 Energy barrier, 117–120
 Entrainment, 298, 315
- Enzyme, 67, 89–90
 purification series, 90
 Ephaptic coupling, 236, 338
 Equilibrium point. *See* fixed point
 Error function. *See* Fitness function
 Erythrocyte, 188
 ES. *See* Evolutionary strategy
 Euler's method, **24–25**, 27
 backward, 25–26
 Event-driven simulation. *See* Discrete-event simulation
 Evolutionary algorithm, 46–49
 Evolutionary strategy, **47–49**, 54, 57–58
 Excitability, 107, 145, 175–176, 267, 271, 273, 326–327
 Excitation, 242–243
 feedback, 325, 350–351
 feedforward, 337
 Excitatory postsynaptic current (EPSC), 141–142
 Excitatory postsynaptic potential (EPSP), 141, 144, 243
 Existence, 2, 26
 Experimental data. *See* Electrophysiological data
 Explicit method. *See* Integration method
 Exploration-exploitation balance, 40–42
 Exponential function, 5, 101
 Exponential peeling, 246–248
 Extracellular space, 338. *See also* Resistance, extracellular
 Eyring theory, 117
- Facilitation. *See* Synaptic facilitation
 Feature-based fitting. *See* Fitness measure
 Feedback, 320, 350–351, 353. *See also* Excitation; Inhibition
 connection, 339, 350
 delayed, 353
 effect of synaptic plasticity, 331
 mechanical, 313–316
 negative, 315
 proprioceptive, 313, 315
 sensory, 313–316
 Fick's law, 81–82
 Field potential, 334, 341
 local, 326, 334, 342
 Field representation
 continuous field model, 322
 mean-field approximation, 323
 Finite-difference method, 24–26, 83
 Finite-element method, 84–85
 Finite-volume method, 84–85, 99, 101
 Firing mode
 repetitive, 15–17, 310
 spontaneous, 279–280, 353
 type I, 175–177
 type II, 174–175, 327
 Firing rate, 164–168, 174–179
 adaptation, **179–180**, 182–184, 325, 353

- model, 322, 350–351
network, 321–323, 329–332, 343–344
Fitness function, 31–37, 40, 45–46, 53–55, 57–58.
See also Penalty function
dynamic, 55
feature-based, 35, 271, 273
multiobjective, 53–54
noisy, 54–55
FitzHugh-Nagumo model, 174–175
Fixation, 188–189, 191–193, 197–198
Fixed point, 6, 7–13, 15–17, 169–171
Flow field, 10–15
Fluorescent dye, 104, 188, 195, 199–200
Flux, 81–83
Frequency tuning, 331
Frequency-intensity curve (*f-I*), 167, 174, 176–179, 184–185, 326–327, 329
Fusion pore, 148–149, 154
- G-protein model, 152
GA. *See* Genetic algorithm
GABA_A receptor, 143–144, 146–147, 150–151. *See also* Inhibition
GABA_B receptor, 152
Gain adaptation, 153, 179
Gain control, 152, 311, 329, 331, 337, 351
Gamma process, 163–164, 337–338
Gap junction, 191, 244, 288, 292, 322, 327, 331–334, 340, 348
Gating. *See* Channel; Current
Gaussian distribution, 85–87
Gear method, 26
Generalization, 271, 345
GIF. *See* Integrate-and-fire neuron model, generalized
Generation, 46–49
GENESIS software, 24, 206, 240, 251, 260–265, 316, 335, 338–341, 348–350, 357, 362
Genetic algorithm, 47, 49, 55, 58
Geometric approach, 9–17
Geometry, 83–84, 87, 238–239
GFP. *See* Green fluorescent protein
GHK equation. *See* Goldman-Hodgkin-Katz equation
Gillespie algorithm, 28, 70, 73–80. *See also* Tau-leaping method
and diffusion, 87–88
direct method, 74–76
first reaction method, 75–76
next reaction method, 76
Glia, 190, 198
Global. *See* Minimum, Parameter fitting
Glutamate, 140, 143–146, 150–152
Glycerol, 190, 192, 195
Goldman-Hodgkin-Katz equation, 93–94, 100, 116, 145, 261
Golgi cell model, 336, 344
Golgi method, 191–192
- Graded transmission, 288, 295, 348
Gradient, 61, 80–88
Grand probability distribution, 69–73
Graph theory, 333
Green fluorescent protein, 193–194
Grid, 83–84, 86–87
- H-current. *See* hyperpolarization-activated current
Half-center oscillator, 287, 292–294, 296–298
Hardware, 147, 305, 347–350
Heart interneuron. *See* Leech
Hebbian learning, 157
Hessian matrix, 39
Heterogeneity, 249, 256, 312, 325, 334–335, 339
Heun's method, 25
Hill-equation, 95
Hines method, 264
Hippocampus, 194, 312
Hodgkin-Huxley equation, 31–32, 108–111, 115, 161, 184, 260, 264, 274, 287, 295
Hodgkin-Huxley model, 108–111, 116, 121, 123, 125–128, 169–170
Homeostatis, 158, 277–278, 283, 325
Homogeneous. *See* Differential equation
Horseradish peroxidase (HRP), 188, 190, 196
Hybrid network, 305, 308–311
Hyperpolarization-activated current, 170, 173, 249, 252, 272, 277–278, 293, 295, 297, 310, 353
- IF. *See* Integrate-and-fire neuron model
IFB. *See* Integrate-and-fire neuron model, or-burst
 I_h . *See* hyperpolarization-activated current
Image stack, 193–194, 200–201, 203
ImageJ software, 201, 203
Immunocytochemistry, 276
Impedance, 168–169, 171, 173, 184
match, 245
Impulse response, 352
Inactivation. *See* Channel
Inactive state, 153
Inferior olive model, 312
Influx. *See* Calcium
Inhibition. *See also* GABA_A receptor, GABA_B receptor
balanced, 281, 325, 337
divisive, 144, 243, 325, 329, 351–352
feedback, 325, 336, 350–351
feedforward, 325–326, 344, 350–351
lateral, 322, 324–325, 351
location, 242–243
shunting (*see* divisive)
subtractive, 325, 329, 352
Initial condition, 2, 10, 14, 20, 90, 279–280
Initial-value problems, 24–26
Inositol triphosphate
diffusion, 99, 103
metabolism, 97
receptor, 96–97

- Input
 colored noise, 177–178
 constant, 165–167, 174, 178–179, 241–242, 244, 343
 oscillatory, 168
 periodic, 174, 338, 352
 relation to output, 318–319
 stochastic, 167–168, 337
 synaptic, 165, 167, 240–241, 280–282
 synchronous, 282
 transient, 241–242, 244
 white noise, 166–167, 174, 177–178, 272–273, 319, 337–338
- Integrate-and-fire neuron model, 161163, **165**, 166–186
 adaptive, **179–180**, 184
 adaptive exponential, 182–183
 exponential, **176–179**, 184
 generalized, **168–174**, 182, 184
 leaky, **165–169**, 184–185, 326
 nonlinear, 175–179
 or-burst, **180–181**, 184
 refractory period, **165**, 167, 176
 quadratic, **176–178**, 182, 326
- Integration method, 24–29. *See also* Diffusion; Reaction
 explicit, **24–26**, 68, 77–78
 implicit, **26**, 68, 78, 264
 order, 25
 step size, **24–27**, 29, 99, 263–264, 340–341, 353
 voltage equation, 264–265
- Interspike interval distribution, 163–164, 167–168, 342
- Invertebrate neuron, 190–191, 274. *See also* Cricket; Leech
- Ion channel. *See* Channel
- Ionic current. *See* Current
- IP_3 . *See* Inositol triphosphate
- ISI distribution. *See* Interspike interval distribution
- Isopotential cell, 237, 247. *See also* Compartment I_t . *See* hyperpolarization-activated current
- Izhikevich model, 182–183
- K^+ channel. *See* Potassium channel
- K_d . *See* Reaction, dissociation constant
- Kinetics. *See also* Channel conductance; Synaptic conductance
 mass-action, 66–69
- Kinetikit software, 357
- λ . *See* Electrotonic, length
- Lamprey swimming, 316
- Langevin equation, 70–71, **78–79**
- Lapicque's model, 161–162
- Lattice, 87
- Leak
 conductance, 278, 296
- current, 109, 260
 reversal potential, 278
- Leaping method. *See* Tau-leaping method
- Learning rule, 156–159
 divergence 157, 158
 rate-based, 157–158
 stability, 159, 331
- Least-square error, 32–33
- Leech
 heart interneuron (HN), 287, 292–299, 309–310
 heart motor neuron, 292
 heartbeat timing network, 292–299
 swimming, 314–316
- Levenberg-Marquardt method, 39
- Light microscopy, 187–196
 measurement error, 202–203
- Limit cycle, **8–9**, 15–17
- Linear congruential generator, 60
- Linear equation. *See* Differential equation
- Linearity, 5–7, 240–241, 244, 252–253, 338, **352**
- Linearization, **8–9**, 12, 15, 169–170
- Lipophilic dye, 192–193
- L-Measure software, 206–207
- L-Neuron software, 213
- Load balance, 347–348
- Long-term depression, **157–158**, 268–270
- Long-term potentiation, 157–159
- LSODE software, 26, 68
- LTD. *See* Long-term depression
- LTP. *See* Long-term potentiation
- Lucifer yellow, 188
- Magnesium
 block, 144–146, 150
 concentration, 145
- MAPK cascade, 62
- Markov model, 28, 108, **121–123**, 124, 126–128, 133, 136, 149–150, 156
- Master equation, **72–74**, 122
- MCell software, 86, 104, 155, 205–206, **358**
- Mean value, 275, 289
- Mechanical model, 313. *See also* Neuromechanical model
- Membrane. *See also* Capacitance; Current; Potential; Resistivity
 -bound molecule, 68
 equation, **1**, 109, 286–288, 321–322
 lipid bilayer, 234–235
 surface, 16, 202, 261, 263, 265, 280
 time constant, 165–167, **237–239**, 243, 246, 337, 350
- Memory model, 152–153, 156–157
- Mersenne twister method, 60
- Mesh, 83–86
- Mesoscopic scale, 69–71
- MesoRD software, 88, **358**
- Metabotropic receptor, 97, 140, 152

- Michaelis-Menten equation, **67**, 96
Microdomain, 100–101, 104
MIEN software, 207–28
Minimum
 global, 40, 55–57
 local, 38, 40, 42, 45–46, 55–57, 254, 257
Mitochondrium, 95
ModelDB database, 137, 273
Modigliani software, 359
Monte Carlo method, 73
MOOSE software, 357, **359–360**
Morphoelectrotrophic transform, 242
Morphology, **187–210**, 247–248, 250, 261, 263, 265, 274, 286–287, 327–328
 artifact, 194–195, 213, 277
 class, 211–212
 correction factor, 202–203
 diversity, 211–212
 parameter, 255
 reconstruction, 187–210, 212–213, 249, 255–258
 symmetry, **214–215**, 218, 221, 227, 230
 tracing, 199, 201
Morris-Lecar model, **12–14**, 22–23, 174–175
Motor pattern generation. *See* Central pattern generator
Multiobjective method, **53–54**, 57
Mutation, **46–49**, 55
- Na⁺ channel. *See* Sodium channel
Na⁺-Ca²⁺ exchanger, 95–96
Nanodomain, 99–101, 104
Navigation modeling, 356
NCS software, 360
Nernst potential, 94, 100, 109, **116**
Nernst-Planck equation, 100
NEST software, 339, 347, **360–362**
Network dynamics, 142, 145, 152, 311, 320–321, 324–325, 329–333, 350–354
Network model, 285–354
 architecture, 323, 354
 design, 318, 323–338
 dilution, 339
 inhibitory, 296–298, 321, 329–332
 large-scale, 317–345, 347–350
 number of neurons, 321, 339
 number of synapses, 320–321, 333, 335–336, 339, 348
 parameter randomization, 324, 333–335
 scale, 328, 339
 semisynthetic, 312
 size, 320, 336–337
 small, 285–299
 stability, 325, 327, 329, 351, 353
 symmetry breaking, 335
Neuron_Morpho software, 201, 207
Neurobiotin, 189–191
neuroConstruct software, 346–347, **361**
NeuroExplorer software, 206
NEUROFIT software, 137
Neurofitter software, 37, 58, 278, **362**
Neurolucida software, 198–201, 206–207
Neuromechanical model, 312–316
 closed-loop, 313–316
 open-loop, 313
NeuroML language, 347
Neuromodulation, 271, 297, 313
NeuroMorpho.Org database, 209, 251, 273
Neuromuscular transform, 313
NEURON software, 24, 137, 206–207, 240, 249, 251, 254–255, 260–261, 264–265, 338, 347–348, 350, **362–363**
Neuron class, 211–212, 324
NeuronDB database, 137
Neuroscience Information Framework (NIF), 208
Neurospaces software, 347
Neurotransmitter. *See* Transmitter
Newton method, 38–39
NLIF. *See* Integrate-and-fire neuron model, nonlinear
NMDA receptor, 140, **144–146**, 149. *See also* Magnesium, block
 calcium influx, **94**, 145
EPSC, 146
model, 150–151
 temperature sensitivity, 145
No-free-lunch theorem, 57–58
Node of Ranvier, 100, 274
Noise, 22–23, 54–55, 69, 335–337. *See also* Input
Nonlinearity, 7–17, 176, 323, 325–327, 329, 338, 352–353
Nonuniqueness, 257, 277–278, **299–305**, 335
Normalization factor, 143
NSL software, 361
Nullcline, 12–17
Numerical methods, 24–29. *See also* Integration method

ODE. *See* Differential equation, ordinary
Ornstein-Uhlenbeck process, 337
Oscillation, 62, 179, 292–299, 311, 327, 329, 333–334, 336, 338, 342. *See also* Half-center oscillator
 axonal delay, 148, 329–332, 353
 damped, 172, 174
 frequency, 315–316, 321, 329–333, 353
 of input, 168, 171, 178
 period, 289–293, 297–299, 310, 316
 phase, 290, 293–294, 297–299, 315–316, 322
Overshoot, 171–174

Parallel processing, 49, **347–349**
Parameter fitting, **31–60**, 256, 288–291, 345. *See also* Fitness function; Minimum; Nonuniqueness
brute force search method, 42
channel kinetics, 111–115, 123–136

- Parameter fitting (cont.)
 chemical model, 88–91
 compartmental model, 256–257, 270–271, 274, 276–278, 280
 global search method, 40–52
 hybrid search method, 55–56, 302
 local search method, 37–40
 Markov model, 133–134
 memorization strategy, 55
 random search method, 44
 stochastic search method, 44–52
 synaptic channel, 159
- Parameter space, 31, 40–42, 56; **289–291**. *See also* Nonuniqueness
- Pareto front, 53–54
- Particles swarm optimization, 49–52
- Partitioning, 347–349
- Parvalbumin, 98, 102
- Passive membrane model, **233–258**, 261–263, 277–278
- Patch-clamp recording
 perforated, 248–249
 whole-cell, 112, 189, 212, 248–249, 270, 276
- Path length, 333
- PDE. *See* Differential equation, partial
- Penalty function, 56
- Periodic trajectory, 8–9, 14–15
- Peristimulus histogram, 282
- Permeability. *See* Channel
- PGENESIS software, 348–349, 357
- Phase. *See also* Oscillation
 locking, 333, 344
 plane, **11–17**, 21, 23, 174–175
 plane trajectory density method, 36–38
 -response curve, 326–327
 space, 8–11
 shift, **178–179**, 315
- Phenomenological model, 153, 156, 159, **265–266**
- Photobleaching, 188
- Pinsky and Rinzel model, 266–267
- PMCA pump, 95–96
- Point neuron model, **161–185**, 265, 285, 326
- Point process, 163–164
 nonstationary, 164
- Poisson process, 22, 28, **163–164**, 166–168, 337, 341, 344
- Poisson distribution, **77–78**, 337
- Population, 46–49, 54–55, 322, 342
- Population vector, **69–72**, 74–75
- Postsynaptic potential, 141, 144, 243
- Potassium channel
 BK-type Ca^{2+} -activated, 101–102, 260
 delayed rectifier, 108–111, 123–124
 SK-type Ca^{2+} -activated, 102
- Potential. *See also* Field Potential; Postsynaptic potential; Voltage
 extracellular, 235, 273
 half-activation, 33, 120
- membrane potential, 1, 109, 165–167, **234**
 reset, 165
 resting, 144, 146, 165, 182, **234**
 reversal, 109, **116**, 144, 146
 threshold, 14, 17, 19, **165**, 167, 176, 179, 182, 184
- Prediction, 1–2, 125–127, 269–270, 291–292, 298–299, 310, 312
- Presynaptic terminal, 104, 139–140, 148–149, 154
- PRNG. *See* Random number generator
- Probability density, 215, 220–221, 230–231
- Probability distribution, 69–74, 77, 85–86
- Propagation, 18–19, 244
- Propensity, 69–78
 function, 69–72
 zero, 72
- Pseudorandom number generator. *See* Random number generator
- PSICS software, 275, **363**
- PSO. *See* Particle swarm optimization
- Pumps, **95–96**, 102–103
 electrogenic, 95–96
- Purkinje cell, 98–99, 102, 250, 262, 280, 300
 model, 263–264, 269–270, 279, 280–281, 305
- Push-pull connectivity, 324–325, 352
- Pyramidal cell, 146, 194, 212, 214, 217–230, 235, 250, 252, 311–312
 model, 17, 184, 266–267, 269, 280, 310
- Python language, 347, 356, 359, 363, 364
- Q10 factor, 130, 145
- QIF. *See* Integrate-and-fire neuron model, quadratic
- Quantal size, 155
- ra*. *See* Resistance, axial
- R_a*. *See* Resistivity, axial
- rand function, 60
- Random number generator, 59–60, 341, 346
- Random walk method, 86–87
- RANDU function, 60
- Raster plot, 341
- Reaction, 61–80
 continuous solution, 79
 deterministic solution method, 66–68
 -diffusion system (*see* Diffusion)
 dimerization, 71–72
 discrete solution, 79
 dissociation constant, 65, 90, 96, 98
 joint probability density function, 73–75
 kinetics, 66–80
 membrane-limited, 68
 mesoscopic rate constant, 69, 71–72
 mixed method, 78–80
 number of molecules, 68–72, 78–79
 numerical integration, 68, 78–80
 rate, 64, 67–68, 117
 second-order, 71
 stochastic, 68–73

- stochastic solution method, 73–76, 79
unimolecular, 71
- Readout neuron, 343
- Rebound depolarization, 327
- Receptor. *See* Synaptic receptor
- Reciprocity, 241
- Recombination operation, 46–48
- Reconstruct software, 203–204
- Reconstruction. *See* Morphology
- Recovery. *See* Channel, inactivation
- Rectification. *See* Goldman-Hodgkin-Katz equation
- Reduced model, 265–266. *See also* Cylinder, equivalent
- Refractory period. *See* Integrate-and-fire neuron model; Poisson process
- Relevance, 345
- Renewal process. *See* Point process
- Repair. *See* Axon, Dendrite
- Repressilator circuit, 62
- Residual, 32–33, 152
- Resistance
- axial, 235–236, 246–250
 - cytoplasmic, 261–263
 - extracellular, 235–237
 - intracellular (*see* axial)
 - input, 234, 236, 239–241, 244–245, 248, 250, 252–253
 - local input, 243
 - membrane, 234–235, 243, 260–261
- Resistivity
- axial, 18, 236, 246, 255
 - membrane, 26–27, 234–235, 243, 246, 248–250, 255–256, 261
- Resonance
- intrinsic, 326–327
 - network, 315, 329, 332, 353
 - stochastic, 337
 - subthreshold, 169, 172, 174, 182–184
- Resonate-and-fire model, 174
- Respiratory pattern generation, 316
- Retina, 353
- rm.* *See* Resistance, membrane
- R_m.* *See* Resistivity, membrane
- R_n.* *See* Resistance, input
- Robustness, 278, 289–290, 305, 345
- Routh-Hurwitz criterion, 7
- Runge-Kutta method, 24–25, 68
- Ryanodine receptor, 97
- Sag, 172, 252
- Sarcoplasmic reticulum, 96
- Saturation, 281, 327
- SBML. *See* Systems Biology Markup Language
- SDE. *See* Differential equation, stochastic
- Search method, 31–60. *See also* Parameter fitting
- Selection, 46–49
- Sensitivity, 257, 290–291, 300
- Sensorimotor model, 315
- SERCA pump, 95–96
- Serial section technique, 198
- Shell, 215–216
- Sharp electrode recording, 212, 248–249
- Sholl, 215–217
- Shooting, 21, 26–27
- Shrinkage, 190–191, 195, 207, 213, 216–218, 255, 274
- Shunt. *See* Soma
- Signaling model, 61–66, 88–91
- Silent synapse, 146
- Simplex algorithm, 39–41
- Simulated annealing, 44–46, 58
- Simulator independence, 347
- SK-type channel. *See* Potassium channel, SK-type Ca^{2+} -activated
- Slice preparation, 154, 188–189, 214, 218
- Small-world topology, 333
- Smoldyn software, 86
- SNNAP software, 363
- Sodium channel
- inactivating, 108–111, 123–127, 287
 - persistent, 172–173, 310
- Solution space, 300–305. *See also* Parameter space connectedness, 304–305
- Soma, 266–267, 269, 274
- dissociated, 276
 - shunt, 248–250
- Space clamp, 146, 242, 275–276
- Space constant, 27, 237, 243–244
- Sparse connectivity, 325, 333, 335,
- Spatial gradient, 66, 80–82
- Spectrogram, 342, 344
- SPICE software, 338
- Spike. *See also* Firing mode; Firing frequency;
- Periodic trajectory
 - conduction, 275, 328
 - dendritic, 265–266
 - initiation, 265, 274
 - response model, 162, 326
 - shape, 270, 296
 - train analysis, 342–343
 - time histogram, 334–342
 - timing-dependent plasticity (*see* Synaptic plasticity)
- Spine. *See* Dendritic spine
- Squid giant axon, 107–108, 234, 236, 274
- strand function, 60
- SSA. *See* Gillespie method
- Stability, 5–10, 12–13, 15, 17, 25–27. *See also* Network model; Synaptic plasticity
- asymptotic, 6–10
 - neutral, 6–7
- Staining, 187–194, 213
- State. *See also* Channel, gating
- closed, 27–28, 119, 123–124
 - open, 22, 27–28, 119–121, 123

- State (cont.)
 resting, 9, 13–14, 16–17, 21, 153
 space, 69, 72–73 (*see also* Phase, space)
STDP. *See* Synaptic plasticity, spike-timing-dependent
Steady state. *See* Fixed point
Steady-state assumption, 66, 264
Steepest-descent method, 39
STEPS software, 88, 104, **364**
Stiff system, 25–26, 29, 68, 264
Stochastic simulation algorithm. *See* Gillespie method
StochSim software, 364
Stoichiometry, 67
Stomatogastric ganglion, 288, 290, 299
Subthreshold dynamics, 14, 168–172. *See also* Resonance
Superposition principle, 352
Surface-to-volume ratio, 94
Susceptibility, 167, 178
SWC format, 201, **206–207**, 210
Swelling, 189, 191, 195
Synapse Model. *See also* Signaling model
 alpha function, 141–142
 difference of two exponentials, 141, **143**
 dynamic, 152–154, 331
 single-exponential decay, 142
 stochastic, 154–156
Synapse Web, 210
Synaptic conductance, 139–159, 281. *See also* Postsynaptic potential; Synapse Model
 decay, 141–143, 153
 decay time constant, 142–143, 145–147
 desensitization, 152
 kinetics, **140–143**, 149–152
 peak of, 141–143
 rise time, **141–143**, 146, 243
 stochastic rate constant, 149–151
 weight, **156–158**, 328, 334–335, 339
Synaptic current 94, **143–144**, 241, 261, 281–282.
See also Excitatory postsynaptic current
Synaptic depression, **152–154**, 325, 331, 353
Synaptic facilitation, **152–153**, 331, 353
Synaptic failure, 325, 327–328
Synaptic input, 243–244, 337
 background, 281–282 (*see also* Inhibition, balanced)
 diffusion approximation, 167
Synaptic latency. *See* Axon, delay
Synaptic noise, 156, 281
Synaptic plasticity, 96, 103, 145, 331. *See also* Synaptic depression; Synaptic facilitation
 long-term, 157–159
 short-term, 152–154
 spike-timing-dependent, 148, **157–158**, 331
Synaptic receptor, 144–147, 149–152, 156
 AMPA (*see* AMPA receptor)
 binding, 149–151
GABA (*see* GABA_A receptor, GABA_B receptor)
metabotropic (*see* Metabotropic receptor)
NMDA (*see* NMDA receptor)
 subunit, 149
Synaptic transmission. *See* Transmitter
Synchronization, 311–312, 333, 335–336, 353. *See also* Oscillation, axonal delay
 artifactual, 340
 measure of, 342–344
 rate covariance, 343–344
Synfire chain model, 326, 350
SYNU software, 205–206
Systems biology, 90–91, 282
Systems Biology Markup Language, 356–358
 τ -leaping method. *See* Tau-leaping method
T-type current. *See* Calcium current
Table, 264–265, 349
Tau-leaping method, 76–79
Temperature, 45–46, 111, 274–275
 dependence, **130**, 145, 159
Temporal filtering, 153
Thalamic relay neuron, 112–114, 127–136, 180–181
Thermodynamic model, **117–121**, 130–133, 136
Theta neuron model, 176–177
Threshold, 54, 157, 352. *See also* Current; Potential
 Time constant, 153, 247, 350. *See also* Channel; Membrane; Synaptic conductance
Time scale, 65, 75, 80, 276–277, 353
Time shift, 34–35, 37
Time step. *See* Integration method, step size
Topology. *See* Connection; Dendrite
Tortuosity, 216
Tracing. *See* Electron tomography; Morphology
Trajectory. *See* Periodic trajectory
Transfer function, 352
Transition probability, 122
Transmitter, 140, 148–152. *See also* Diffusion
 release site, 154–155
 reuptake, 149
 transporter, 149, 152, 155
 unbinding, 142–143, 146
 vesicle, 140, 148–149, 152, 154–155
 vesicle depletion, 152–153
Trapezoidal rule, 25
Type I/II mode. *See* Firing mode
Ultrastructural level. *See* Electron microscopy
Uniqueness, 2, 5, 345. *See also* Nonuniqueness
Unit, 236, 277
 extensive, 68
 intensive, 68
Valence. *See* Charge
Validation, 227, 229, 271, 278–279, 340–342, 345, 349

- Variability, 289–290, 299. *See also* Nonuniqueness
synaptic, 149, 154–156
- Varicosity, 189, 194, 200, 202
- VCELL software, 365
- Vesicle. *See* Transmitter
- VIAS software, 201
- VLSI, 147, 154
- V_m . *See* Potential, membrane
- V_t . *See* Potential, threshold
- Voltage. *See also* Potential; Sag
attenuation 241–245, 280
clamp, 108, 111–114, 125–127, 242, 275, 281 (*see also* Space clamp)
decay, 237–239, 243, 247
dependence (*see* Channel, kinetics)
- Volume, 68–69, 71, 94, 101
- Voxel, 83–84, 87–88
- Wave propagation, 19–21, 315, 322, 333, 335
- Well-mixed system, 66–68, 87
- Wiener process, 27
- Winner-takes-all model, 325, 351
- XPPAUTO software, 12, 27, 316, 323, 365–366
- Xvotrace software, 205

