The NEURON Book

N.T. Carnevale[1] and M.L. Hines[2]
Departments of [1]Psychology and [2]Computer Science
Yale University, New Haven, CT
ted.carnevale@yale.edu
michael.hines@yale.edu

Who should read this book

This book is about how to use the NEURON simulation environment to construct and apply empirically−based models of neurons and neural networks. It is written primarily for neuroscience investigators, teachers, and students, but readers with a background in the physical sciences or mathematics who have some knowledge about brain cells and circuits and are interested in computational modeling will also find it helpful. The emphasis is on the most productive use of NEURON as a means for testing hypotheses that are founded on experimental observations, and for exploring ideas that may lead to the design of new experiments. Therefore the book uses a problem−solving approach, with many working examples that readers can try for themselves.

What this book is, and is not, about

Formulating a *conceptual model* is an attempt to capture the essential features that underlie some particular function. This necessarily involves simplification and abstraction of real−world complexities. Even so, one may not necessarily understand all implications of the conceptual model. To evaluate a conceptual model it is often necessary to devise a hypothesis or test in which the behavior of the model is compared against a prediction. *Computational models* are useful for performing such tests. The conceptual model and the hypothesis should determine what is included in a computational model and what is left out. This book is not about how to come up with conceptual models or hypotheses, but instead focuses on how to use NEURON to create and use computational models as a means for evaluating conceptual models.

What to read, and why

The first chapter conveys a basic idea of NEURON's primary domain of application by guiding the reader through the construction and use of a model neuron. This exercise is based entirely on NEURON's GUI, and requires no programming ability or prior experience with NEURON whatsoever.

The second chapter considers the role of computational modeling in neuroscience research from a general perspective. Chapters 3 and 4 focus on aspects of applied mathematics and numerical methods that are particularly relevant to computational neuroscience. Chapter 5 discusses the concepts and strategies that are used in NEURON to simplify the task of representing neurons, which (at least at the level of synapses and cells) are distributed and continuous in space and time, in a digital computer, where

neither time nor numeric values are continuous. Chapter 6 returns to the topic of model construction, emphasizing the use of programming.

Chapters 7 and 8 provide "inside information" about NEURON's standard run and initialization systems, so that readers can make best use of their features and customize them to meet special modeling needs. Chapter 9 shows how to use the NMODL programming language to add new biophysical mechanisms to NEURON. This theme continues in Chapter 10, which starts with mechanisms of communication between cells (gap junctions, graded and spike−triggered synaptic transmission), and moves on to models of artificial spiking neurons (e.g. integrate and fire cells). The first half of Chapter 11 is a tutorial on NEURON's GUI tools for creating simple network models, and the second half shows how to use the strengths of the GUI and `hoc` programming to create more complex networks.

Chapter 12 discusses the elementary features of the `hoc` programming language itself. Chapter 13 describes the object−oriented extensions that have been added to `hoc`. These extensions have greatly facilitated construction of NEURON's GUI tools, and they can also be very helpful in many other complex programming tasks such as creating and managing network models. Chapter 14 presents an example of how to use object oriented programming to increase the functionality of NEURON.

Table of contents

Note: page numbers in each chapter start from 1.

Chapter 1. A tour of the NEURON simulation environment

Chapter 5. Representing neurons with a digital computer

Chapter 6. How to build and use models of individual cells

Chapter 7. How to control simulations

Chapter 8. How to initialize simulations

Chapter 9. How to expand NEURON's library of mechanisms

Chapter 10. Synaptic transmission and artificial spiking cells

Chapter 11. Modeling networks

Note: this chapter is in an early draft, so we present its outline.

Building a simple network with the GUI

Conceptual model of recurrent inhibition:

Motoneuron with excitatory afferent and Renshaw cell

Design of the computational model:

Represent the motoneuron with a biophysical model

Represent the afferent spike train and Renshaw cell with artificial cell models

Use the event delivery system to represent axonal conduction delays and synaptic latency

Implementing the computational model

Specifying the motoneuron model: the NetReadyCellGUI tool

Specifying the afferent spike train and the Renshaw cell: the ArtCellGUI tool

Specifying the network connections and creating the net: the NetGUI tool (Network Builder)

Running a simulation and plotting spike trains

Combining the GUI and programming to build a complex network model

Conceptual model of inhibitory synchronization

The cells: spontaneously spiking neurons with a range of natural firing frequencies

Chapter 12. hoc −− NEURON's interpreter

Chapter 13. Object−oriented programming

Chapter 14. How to modify NEURON itself

