CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

1
A tour of the NEURON simulation environment

. . . so, entering, the first thing I did was to stumble over an ash-box
in the porch. Ha! thought I, ha, as the flying particles almost choked me,
are these ashes from that destroyed city, Gomorrah?

1.1 Modeling and understanding

Modeling can have many uses, but its principal benefit is to improve understanding. The chief question that it addresses is whether what is known about a system can account for the behavior of the system. An indispensable step in modeling is to postulate a *conceptual model* that expresses what we know, or think we know, about a system, while omitting unnecessary details. This requires considerable judgment and is always vulnerable to hindsight and revision, but it is important to keep things as simple as possible. The choice of what to include and what to leave out depends strongly on the hypothesis that we are studying. The issue of how to make such decisions is outside the primary focus of this book, although from time to time we may return to it briefly.

The task of building a *computational model* should only begin after a conceptual model has been proposed. In building a computational model we struggle to establish a match between the conceptual model and its computational representation, always asking the question: would the conceptual model behave like the simulation? If not, where are the errors? If so, how can we use NEURON to help understand why the conceptual model implies that behavior?

1.2 Introducing NEURON

NEURON is a simulation environment for models of individual neurons and networks of neurons that are closely linked to experimental data. NEURON provides numerically sound, computationally efficient tools for conveniently constructing, exercising, and managing models, so that special expertise in numerical methods or programming is not required for its productive use. Increasing numbers of experimentalists and theoreticians are incorporating it into their research strategies. As of this writing, more than 460 scientific

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

publications have reported work done with NEURON on topics that range from the molecular biology of voltage-gated channels to the operation of networks containing thousands of neurons (see **Research reports that have used NEURON** at `http://www.neuron.yale.edu/neuron/bib/usednrn.html`).

In the following pages we introduce NEURON by going through the development of a simple model from start to finish. This will require us to perform each of these tasks:

1. State the question that we are interested in
2. Formulate a conceptual model
3. Implement the model in NEURON
4. Instrument the model, that is, attach signal sources and set up graphs
5. Set up controls for running simulations
6. Save the model with instrumentation and run controls
7. Run simulation experiments
8. Analyze results.

Since our aim is to provide an overview, we have chosen a simple model that illustrates just one of the NEURON's strengths: the convenient representation of the spread of electrical signals in a branched dendritic architecture. We could do this by writing instructions in NEURON's programming language `hoc`, but for this example we will employ some of the tools that are provided by its graphical user interface (GUI). Later chapters examine `hoc` and the graphical tools for constructing models and managing simulations in more detail, as well as many other features and applications of the NEURON simulation environment (e.g. complex biophysical mechanisms, neural networks, customization of the user interface).

1.3 State the question

The scientific issue that motivates the design and construction of this model is the question of how synaptic efficacy is affected by synaptic location and the anatomical and biophysical properties of the postsynaptic cell. This has been the subject of too many experimental and theoretical studies to reference here. Interested readers will find numerous relevant publications in NEURON's on-line bibliography (cited above), and may retrieve working code for several of these from ModelDB (`http://senselab.med.yale.edu/senselab/modeldb/`).

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

1.4 Formulate a conceptual model

Most neurons have many branches with irregularly varying diameters and lengths (Fig. 1.1 A), and their membranes are populated with a wide assortment of ionic channels that have different ionic specificities, kinetics, dependence on voltage and second messengers, and spatial distributions. Scattered over the surface of the cell may be hundreds or thousands of synapses, some with a direct effect on ionic conductances (which may also be voltage-dependent) while others act through second messengers. Synapses themselves are far from simple, often displaying stochastic and use-dependent phenomena that can be

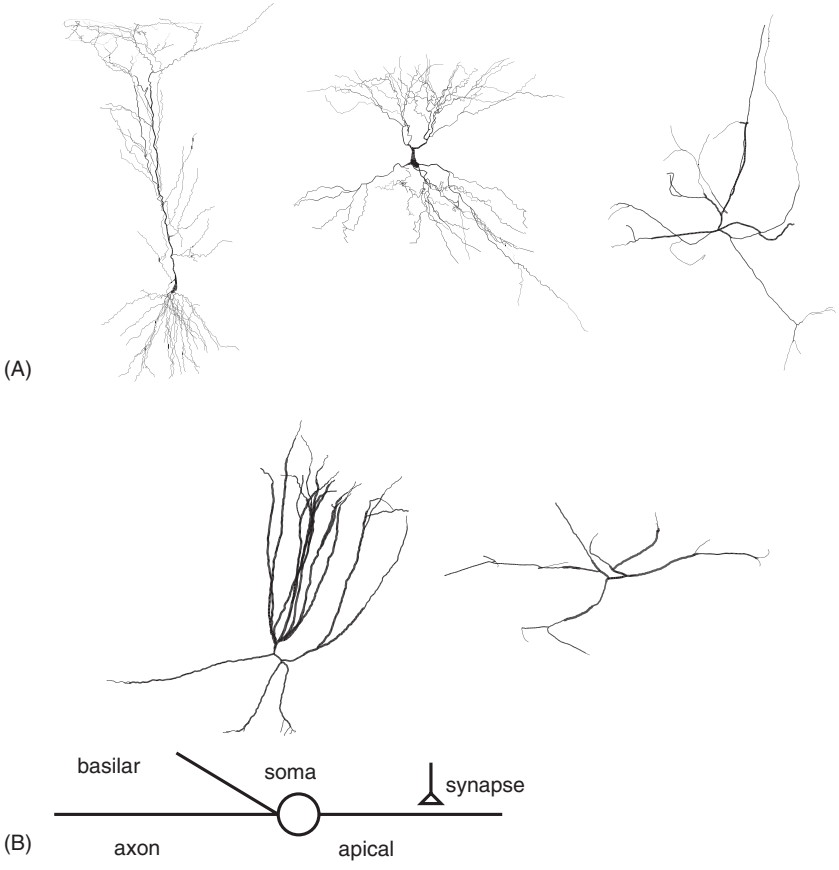


Figure 1.1. (A) Clockwise from top left: Ca1 and Ca3 pyramidal neurons (from D.A. Turner); calbindin-, parvalbumin-, and calretinin-positive interneurons (from A.I. Gulyás). (B) Our conceptual model neuron. The conductance change synapse can be located anywhere on the cell.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

Table 1.1. *Model cell parameters*

	Length (μm)	Diameter (μm)	Biophysics
Soma	30	30	HH g_{Na}, g_K, and g_{leak}
Apical dendrite	600	1	Passive with $R_m = 5000\ \Omega\ cm^2$, $E_{pas} = -65\ mV$
Basilar dendrite	200	2	Same as apical dendrite
Axon	1000	1	Same as soma

$C_m = 1\ \mu f/cm^2$; cytoplasmic resistivity $= 100\ \Omega cm$; temperature $= 6.3°C$.

quite prominent, and frequently being subject to various pre- and postsynaptic modulatory effects. Given all this complexity, we might well ask if it is possible to understand anything without understanding everything. From the very onset we are forced to decide what to include and what to omit.

Suppose we are already familiar with the predictions of the basic ball and stick model (Rall 1977; Jack et al. 1983), and that experimental observations motivate us to ask questions such as: How do synaptic responses observed at the soma vary with synaptic location if dendrites of different diameters and lengths are attached to the soma? What happens if some parts of the cell have active currents, while others are passive? What if a neuromodulator or shift of the background level of synaptic input changes membrane conductance?

Then our conceptual model might be similar to the one shown in Fig. 1.1 B. This model includes a neuron with a soma that gives rise to an axon and two dendritic trunks, and a single excitatory synapse that may be located at any point on the cell.

Although deliberately more complex than the prototypical ball and stick, the anatomical and biophysical properties of our model are much simpler than the biological original (Table 1.1). The axon and dendrites are simple cylinders, with uniform diameters and membrane properties along their lengths. The dendrites are passive, while the soma and axon have Hodgkin–Huxley (HH) sodium, potassium, and leak currents, and are capable of generating action potentials (Hodgkin and Huxley 1952). A single synaptic activation causes a localized transient conductance increase with a time course described by an alpha function

$$g_s(t) = \begin{cases} 0 & \text{for } t < t_{act} \\ g_{max} \dfrac{(t - t_{act})}{\tau_s} \, e^{-\frac{(t - t_{act})}{\tau_s}} & \text{for } t \geq t_{act} \end{cases} \tag{1.1}$$

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

Table 1.2. *Synaptic mechanism parameters*

g_{max}	0.05 μS
τ_s	0.1 ms
E_s	0 mV

where t_{act} is the time of synaptic activation, and g_s reaches a peak value of g_{max} at $t = \tau_s$ (see Table 1.2 for parameter values). This conductance increase mechanism is just slightly more complex than the ideal current sources used in many theoretical studies (Rall 1977; Jack et al. 1983), but it is still only a pale imitation of any real synapse (Bliss and Lømo 1973; Ito 1989; Castro-Alamancos and Connors 1997; Thomson and Deuchars 1997).

1.5 Implement the model in NEURON

With a clear picture of our model in mind, we are ready to express it in the form of a computational model. Instead of writing instructions in NEURON's programming language `hoc`, for this example we will employ some of the tools that are provided by NEURON's GUI.

We begin with the CellBuilder, a graphical tool for constructing and managing models of individual neurons. At this stage, we are not considering synapses, stimulating electrodes, or simulation controls. Instead we are focussing on creating a representation of the continuous properties of the cell. Even if we were not using the CellBuilder but instead were developing our model entirely with `hoc` code, it would probably be best for us to follow a similar approach, that is, specify the biological attributes of the model cell separately from the specification of the instrumentation and control code that we will use to exercise the model. This is an example of modular programming, which is related to the "divide and conquer" strategy of breaking a large and complex problem into smaller, more tractable steps.

The CellBuilder makes it easier for us to create a model of a neuron by allowing us to specify its architecture and biophysical properties through a graphical interface. When we are satisfied with the specification, the CellBuilder will generate the corresponding `hoc` code for us. Once we have a model cell, we will be ready to use other graphical tools to attach a synapse to it and plot simulation results (see **1.6 Instrument the model**).

The images in the following discussion were obtained under MSWindows; the appearance of NEURON under UNIX, Linux, and MacOS is quite similar.

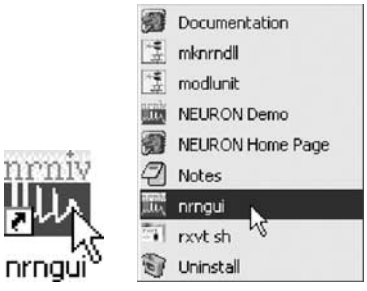
© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

1.5.1 Starting and stopping NEURON

No matter what a program does, the first thing you have to learn is how to start and stop it. To start NEURON under UNIX or Linux, just type `nrngui` on the command line and skip the remainder of this paragraph. Under MSWindows, double click on the nrngui icon on your desktop (Fig. 1.2A); if you don't see one there, bring up the NEURON program group (i.e. use Start / Program Files / NEURON) and select the `nrngui` item (Fig. 1.2B). If you are using MacOS, open the folder where you installed NEURON and double click on the nrngui icon.



(A) (B)

Figure 1.2. Under MSWindows, start NEURON by clicking on the nrngui icon on the desktop (A) or selecting the nrngui item in the NEURON program group (B).

You should now see the NEURON Main Menu (Fig. 1.3A), which offers a set of menus for bringing up graphical tools for creating models and running simulations. If you are using UNIX or Linux, a "banner" that includes the version of NEURON you are running will be printed in the xterm where you typed `nrngui`, and the prompt will change to `oc>` to indicate that NEURON's `hoc` interpreter is running. Under MacOS and MSWindows, the "banner" and `oc>` prompt will appear in a new console window (Fig. 1.3B).

There are three different ways to exit NEURON; you can use whichever is most convenient:

1. type `^D` (i.e. control D) at the `oc>` prompt
2. type `quit()` at the `oc>` prompt
3. click on File in the NEURON Main Menu, scroll down to Quit, and release the mouse button (Fig. 1.4).

1.5.2 Bringing up a CellBuilder

To get a CellBuilder just click on Build in the NEURON Main Menu, scroll down to the CellBuilder item, and release the mouse button (Fig. 1.5).

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

(A)

(B)

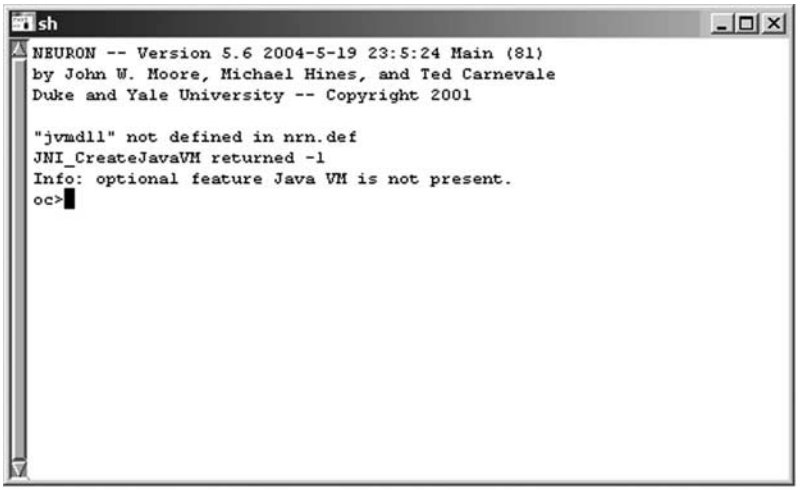


Figure 1.3. (A) The NEURON Main Menu toolbar. (B) NEURON's "banner" and oc> prompt in an MSWindows console window.

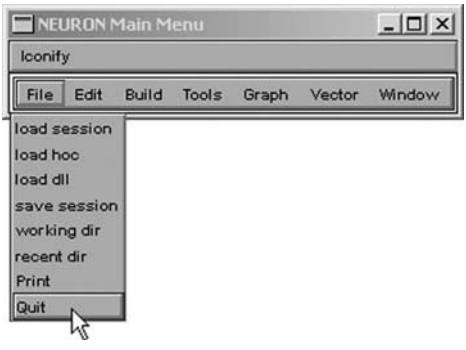


Figure 1.4. One way to exit NEURON is to click on File / Quit in the NEURON Main Menu toolbar.

Across the top of the CellBuilder is a row of radio buttons and a checkbox, which correspond to the sequence of steps involved in building a model cell (Fig. 1.6). Each radio button brings up a different page of the CellBuilder, and each page provides a view of the model plus a graphical interface for defining properties of the model. The first four pages (Topology, Subsets, Geometry,

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

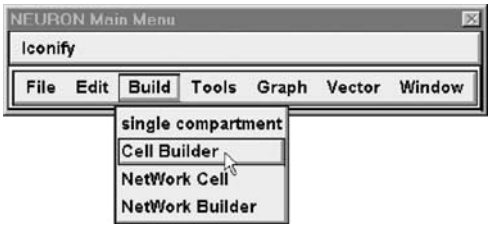


Figure 1.5. Using the NEURON Main Menu to bring up a CellBuilder.

Figure 1.6. Top panel of the CellBuilder.

Biophysics) are used to create a complete specification of a model cell. On the Topology page, we will set up the branched architecture of the model and give a name to each branch, without regard to diameter, length, or biophysical properties. We will deal with length and diameter on the Geometry page, and the Biophysics page is where we will define the properties of the membrane and cytoplasm of each of the branches.

The Subsets page deserves special comment. In almost every model that has multiple branches, two or more of them will have at least some biophysical attributes that are identical, and there are often significant anatomical similarities as well. Furthermore, we can almost always apply the d_lambda rule for compartmentalization throughout the entire cell (see below). We can take advantages of such regularities by assigning shared properties to several branches at once. The Subsets page is where we group branches into subsets, on the basis of shared features, with an eye to exploiting these commonalities on the Geometry and Biophysics pages. This allows us to create a model specification that is compact, efficient, and easily understood.

1.5.3 Entering the specifications of the model cell

1.5.3.1 Topology

We start by using the Topology page to set up the branched architecture of the model. As Fig. 1.7 shows, when a new CellBuilder is created, it already contains a branch (or "section," as it is called in NEURON) that will serve as the root of the branched architecture of the model (the root of a tree is the branch that has no parent). This root section is initially called "soma," but we can rename it if we desire (see below).

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

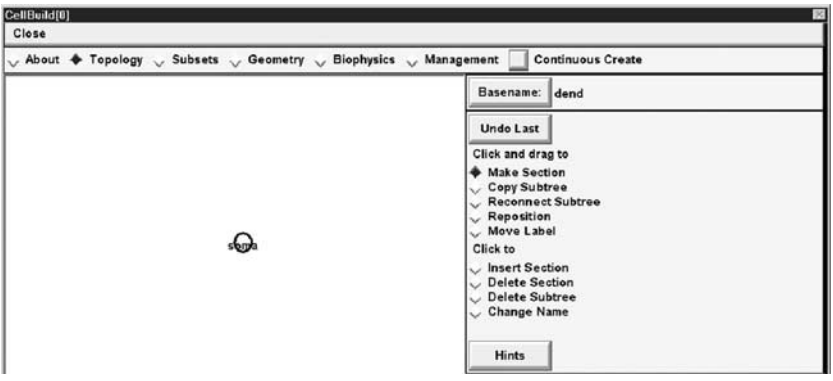


Figure 1.7. The Topology page. The left panel shows a simple diagram of the model, which is called a "shape plot." The right panel contains many functions for editing the branched architecture of a model cell.

The Topology page offers many functions for creating and editing individual sections and subtrees. We can make the section that will become our apical dendrite by following the steps presented in Fig. 1.8. Repeating these actions a couple more times (and resorting to functions like Undo Last, Reposition, and Delete Section as needed to correct mistakes) gives us the basilar dendrite and axon.

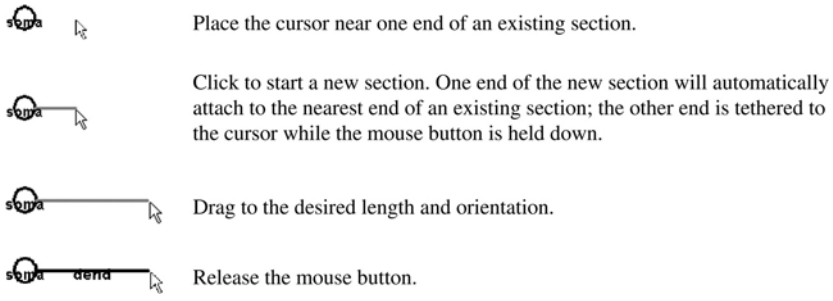


Figure 1.8. Making a new section. Verify that the Make Section radio button is on, and then perform with the mentioned steps.

Our model cell should now look like Fig. 1.9. At this point some minor changes would improve its appearance: moving the labels away from the sections so they

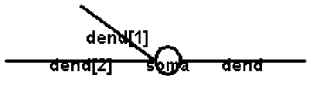


Figure 1.9. The model after all sections have been created.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
0521843219 - The Neuron Book
Ted Carnevale and Michael Hines
Excerpt
More information

are easier to read (Fig. 1.10), and then renaming the apical and basilar dendrites and the axon (Figs. 1.11 and 1.12). The final result should resemble Fig. 1.13.

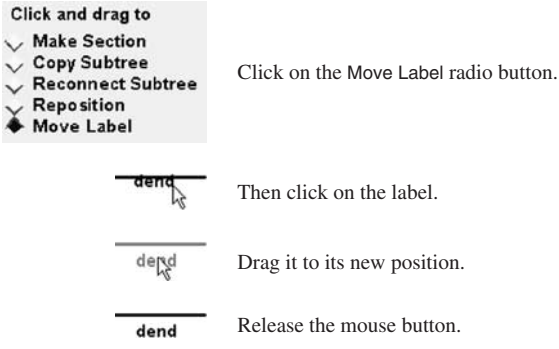


Figure 1.10. How to change the location of a label.

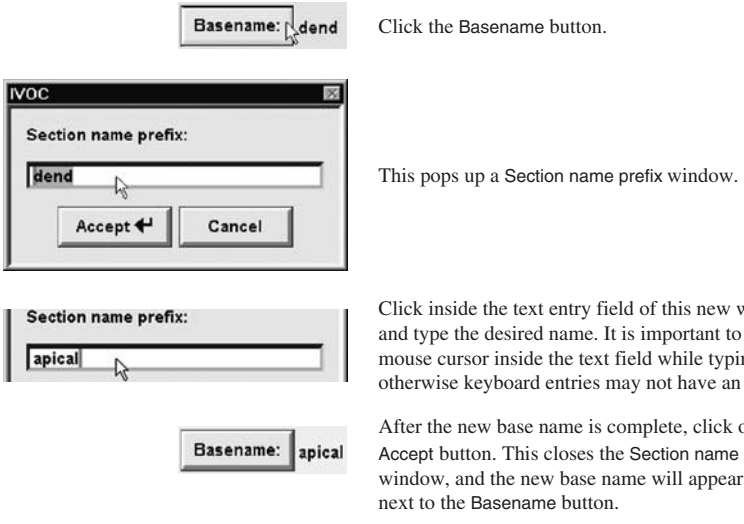


Figure 1.11. Preparing to change the name of a section. Each section we created was automatically given a name based on "dend." To change these names, we must first change the base name as shown here.

1.5.3.2 Subsets

As mentioned above, the Subsets page (Fig. 1.14) is for grouping sections that share common features. Well-chosen subsets can save a lot of effort later by helping us create very compact specifications of anatomical and biophysical properties.

© Cambridge University Press www.cambridge.org