
Chapter 2

Compartmental Modeling

JAMES M. BOWER and DAVID BEEMAN

Before beginning to explore the tutorials, it is important to understand something about the assumptions and the mathematical models that underlie these simulations. Thus, although the first section of the book describes what are essentially “point and click” tutorials, it is important not to use these tutorials blindly. Their effective use requires some understanding of the basics of neural modeling, as well as the concepts in neuroscience that are introduced along with the tutorials. Entire books have been written on this subject, so obviously we can only highlight the issues here. However, throughout the text we have referenced other sources of information. If you are seriously considering building models yourself, we would strongly recommend that you consult these references.

2.1 Modeling Neurons

Figure 2.1A shows an example neuron based on a drawing of a pyramidal cell by Ramón y Cajal that we would like to model, either as a single cell, or as a component in a network of interacting neurons. This figure shows the tree-like structure of the dendrites, which receive synaptic inputs from other neurons. Synaptically activated ion channels in the dendrites create postsynaptic potentials that, we assume here for simplicity, are passively propagated to the pyramid-shaped cell body (soma) where voltage-activated ion channels may create action potentials. In most cells, these channels are concentrated near the base of the soma in the region called the *axon hillock* near the axon. The long axon at the bottom of the figure propagates action potentials to terminal branches that form synapses with other neurons. In some cases (Chapter 7) neurons may have voltage-activated channels in their dendrites.

This not only complicates their electrical properties and thus their simulation, but also is responsible for the complex dynamics of these neurons.

2.1.1 Detailed Compartmental Models

When constructing detailed neuronal models that explicitly consider all of the potential complexities of a cell, the increasingly standard approach is to divide the neuron into a finite number of interconnected anatomical compartments. Figure 2.1B shows a simplified model in which the neuron is divided into several dendrite compartments, a soma, and an axon. Each compartment is then modeled with equations describing an equivalent electrical circuit (Rall 1959). With the appropriate differential equations for each compartment, we can model the behavior of each compartment as well as its interactions with neighboring compartments.

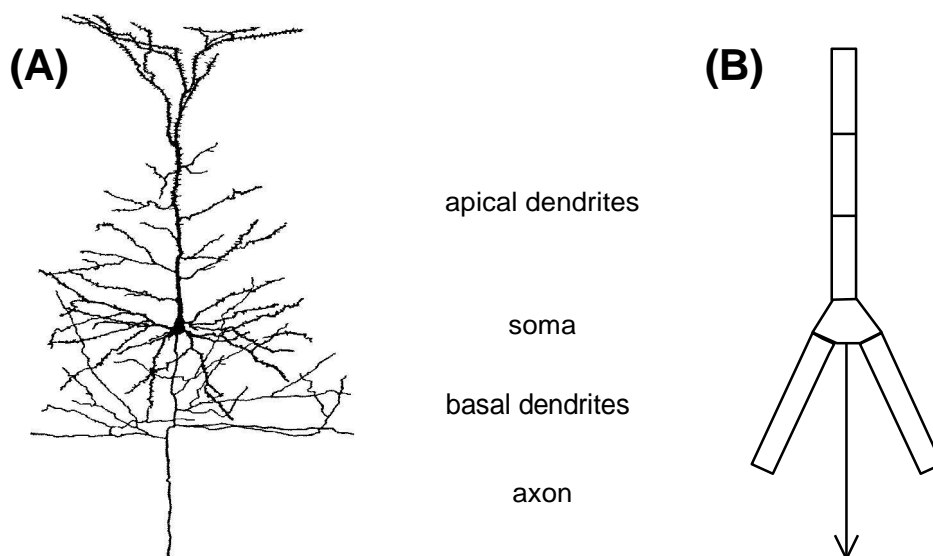


Figure 2.1 (A) A pyramidal cell with dendrites, soma, and axon. (B) A simplified discrete compartmental model of the same neuron.

In this type of detailed compartmental model, each compartment must be made small enough to be at approximately the same electrical potential. Often this means constructing simulations out of very large numbers of compartments. For example, we have recently published a GENESIS model of a cerebellar Purkinje cell that uses 4550 compartments and 8021 channels (De Schutter and Bower 1994a,b). The representation of this model is shown in Fig. 2.2.

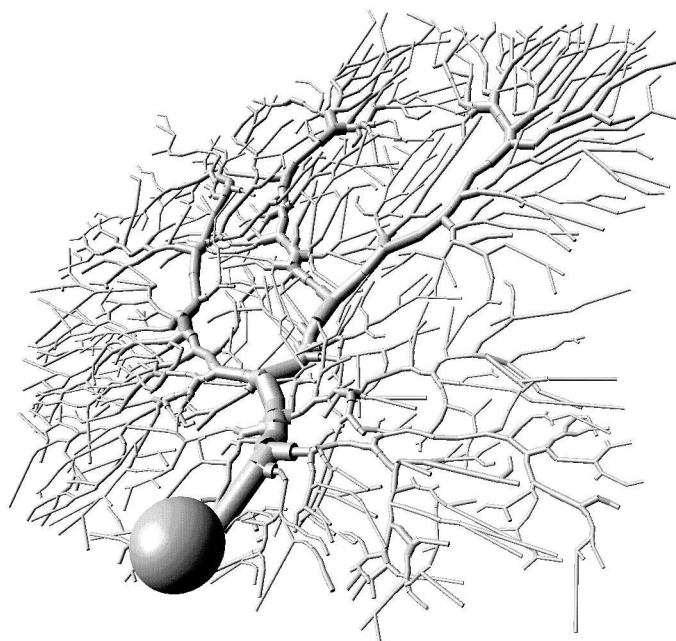


Figure 2.2 A detailed multi-compartmental model of a cerebellar Purkinje cell, created with GENESIS by De Schutter and Bower (1994a,b). Visualization by Jason Leigh using the GENESIS Visualizer program. The experimental data describing the cell morphology was provided by M. Rapp, I. Segev and Y. Yarom.

2.1.2 Equivalent Cylinder Models

For some purposes, it may be adequate to model neurons with a smaller number of non-equipotential compartments. Models of this sort can be used to model basic electrical properties of cells, or to construct small networks of neurons. Under these conditions, there are defined methods for constructing neuronal models dependent on the anatomy and physiology of the neuron in question. Many of these have been pioneered by Wilfrid Rall (cf., Segev, Rinzel and Shepherd 1995). For example, Rall has shown analytically that if dendritic trees approximately satisfy a “ $3/2$ power law” and do not contain active conductances, they can safely be mapped into an equivalent linear structure (Chapter 5). In this way, under defined conditions, a complicated branching structure can be approximated by a much simpler linear dendrite model, as was done in the model shown in Fig. 2.1B. However, in general, as the known complexity of the physiology or anatomy of the neuron increases, it is usually necessary to revert to full-blown compartmental models.

2.1.3 Single and Few Compartment Models

In cases where large numbers of neurons are being placed in network models, limited computer resources sometimes require that neurons be modeled with single compartments or a very small number of compartments. For example, a large-scale GENESIS simulation of the olfactory cortex uses a network of 4500 neurons containing simple model pyramidal cells similar to the one shown in Fig. 2.1B (Wilson and Bower 1989, 1992). As you will see for yourself in Chapter 9, even such simplified neurons can sometimes capture experimentally observed behavior. In Chapter 7, we will see that only a single compartment is needed to model the behavior of some invertebrate “pacemaker” neurons. On the other hand, when building models of this type, one must always be aware that there are many local “computations” that occur in the extensive dendritic system of many neurons. Again, if these are of interest to the modeler, it is usually necessary to use hundreds or thousands of compartments.

2.2 Equivalent Circuit of a Single Compartment

Having described the general approaches to modeling single neurons, we now discuss in a bit more detail the basis for compartmental modeling. The reader should note that this section is intended as a very basic overview of neural modeling. The topics covered here are treated in more detail in Chapters 4–6 and by Segev, Fleshman and Burke (1989).

As we have described, the notion of an equivalent electrical circuit for a small piece of cellular membrane is the basis for all compartmental modeling. This arises from the fact that neuronal membranes have been demonstrated to behave as simple electrical circuits with some capacitance, resistance, and voltage sources. These model parameters define the so-called *passive properties* that are responsible for the way that electrical impulses are transmitted along the dendritic tree. It is generally necessary, as well as advisable, to begin all single cell modeling efforts with a consideration of passive cellular properties of the cell. These properties form the basis for the usually more interesting neuronal behavior that arises from the *active properties* provided by different voltage- or ligand-dependent conductances. If the passive properties are not modeled correctly, spurious results with active conductances are likely to be obtained.

Figure 2.3 shows the equivalent electrical circuit of a basic neural compartment. Here, V_m represents the membrane potential, or the potential in the interior of a compartment relative to a point outside the cell. The “ground” symbol at the bottom of the figure represents this external point, taken to be at zero potential. As the conducting ionic solutions inside and outside of the cell are separated by the cell membrane, the compartment acts as a capacitor. This is charged or discharged by current flowing into or out of the compartment. This current flow may be from adjacent compartments, from the passage of ions through channels in the cell membrane, or from current injection from an electrode inserted into the

cell. The membrane potential appears across the membrane capacitance C_m , and can cause a current flow into or out of the compartment at the left through the axial resistance R_a when there is a difference in potential $V_m - V_m''$ between the two compartments. Likewise, there may be a flow of current into or out of the primed compartment at the right through its axial resistance R_a' .

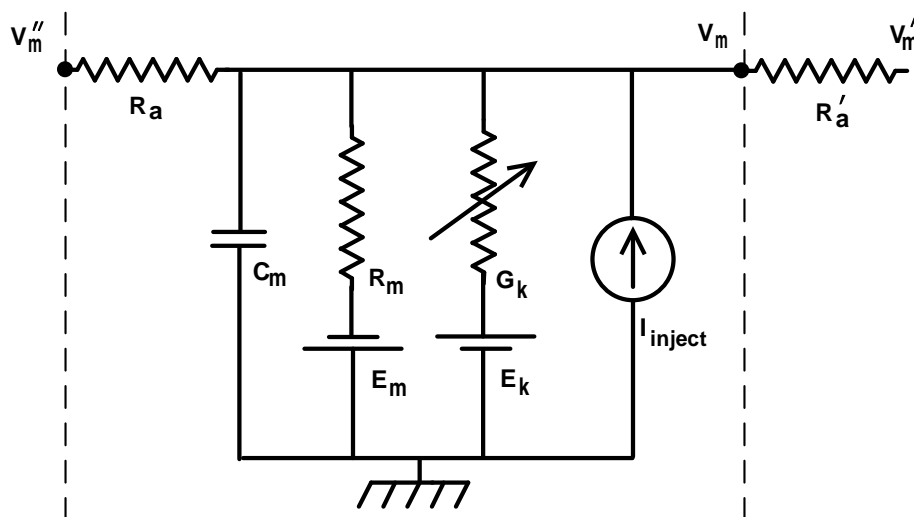


Figure 2.3 The equivalent circuit for a “generic” neural compartment.

The resistor with the arrow through it represents one of many possible variable channel conductances that are specific to a particular ion or combination of ions that give individual neurons and neuron types their unique computational properties. By convention, these are described in terms of the conductance G_k rather than the resistance. As the conductance is the reciprocal of resistance, the units of G_k are in reciprocal ohms, or siemens. Differences in the concentration of the ion between the inside and the outside of the cell result in an osmotic pressure which tends to move ions along the concentration gradient. The resulting charge displacement creates a potential difference that opposes this flow. The membrane potential at which there is no net flux of the ion is the *equilibrium potential* (or *reversal potential*) E_k , represented by a battery in series with the conductance. In the absence of synaptic input, current injection, or spontaneous firing of action potentials, V_m will approach a steady state *rest potential* E_{rest} , typically in the range of -40 to -100 mV. This is determined by the condition that there is no net current flow into the cell from the various types of ion channels.

The other resistor and battery linking the exterior and the interior of the cell represent the combined effect of passive channels (mainly those for chloride ions) having a relatively fixed conductance. The resistance is usually called the *membrane resistance* R_m , although

it is sometimes referred to as a *leakage conductance* $G_{leak} = 1/R_m$. The associated equilibrium potential E_m is typically close to the rest potential. In some cases, it is given a slightly different value, E_{leak} , in order to reduce the net channel current to zero when $V_m = E_{rest}$. Finally, the current source I_{inject} represents an optional injection current which could be provided by an electrode inserted into the compartment.

One may then calculate V_m using a differential equation which expresses the fact that the rate of change of the potential across C_m is proportional to the net current flowing into the compartment to charge the capacitance. On the right-hand side of Eq. 2.1, Ohm's law is used to calculate the current due to each of the sources shown in Fig. 2.3:

$$C_m \frac{dV_m}{dt} = \frac{(E_m - V_m)}{R_m} + \sum_k [(E_k - V_m)G_k] + \frac{(V_m' - V_m)}{R_a'} + \frac{(V_m'' - V_m)}{R_a''} + I_{inject}. \quad (2.1)$$

Here, the sum over k represents a sum over the different types of ion channels that are present in the compartment. The sign convention used in the GENESIS simulator defines a positive channel current to be one that causes a flow of positive charge *into* the compartment. The variable conductance of each channel type G_k gives the net effect of many individual channels that open and close in a binary manner.

To model this on a computer, we need to numerically solve Eq. 2.1 for each compartment. Of course, the V_m'' and V_m' in the adjacent compartments affect the currents flowing into or out of the compartments, so we are solving many coupled equations in parallel. Also, we will need good models for the way that the conductances vary with voltage, time or synaptic input.

2.3 Axonal Connections, Synapses and Networks

Typically, but not always, neurons communicate by means of chemical synapses. The most common situation is one in which an action potential causes the release of a neurotransmitter from a presynaptic terminal at the end of an axon branch. This diffuses across a narrow gap to the postsynaptic junction (usually on a dendrite), causing an increase in conductance for a specific set of ion channels that are sensitive to this transmitter. However, synaptic connections may also be found between two axons or between two dendrites. In many cases, axons make connections to the cell body, rather than to dendritic branches.

Usually, we can treat an axon as a simple delay line for the propagation of action potentials, although it could also be modeled as a series of compartments if we were interested in understanding the details of axonal propagation. In most cases, the change in the postsynaptic channel conductance is a simple function of time which may be determined from experimental measurements. Then, we may avoid having to model the details of the process of presynaptic transmitter release, its binding to postsynaptic receptors, and the way

in which the permeability of the postsynaptic membrane is affected. Instead, we may use an analytical expression such as the *alpha function*, described in Chapter 6, to represent the resulting conductance of synaptically activated channels. When more detailed models of the biochemical reactions underlying synaptic transmission are required, they may be created using the techniques described in Chapter 10 and the graphical interface *Kinetikit*, which was created for the development of models of biochemical signaling pathways.

Of course, each synapse has associated with it an effect of a particular magnitude or “weight” on the postsynaptic cell. Furthermore, these weights can change under some circumstances. The implementation of this *synaptic plasticity* is discussed in Chapter 15. Chapter 19 treats one class of receptors, the voltage-dependent NMDA receptors, which have been shown to confer such weight-changing properties on synapses. Chapter 19 also considers electrical synapses, which are yet another type of connection between cells. Once we have modeled single neurons and the ways in which they may interact, we can proceed to modeling neural circuits and networks. Example models are discussed in Chapters 8 and 9, and the details of constructing your own network simulations are given in Chapter 18.

2.4 Simulation Accuracy

Once a modeler has constructed a simulation, the accuracy of the results depends on many factors, from the quality of the data used to construct the simulation, to the way in which the simulation is run numerically. In the case of an analytic solution, one knows the result is correct, assuming that the underlying model is correct. For numerical simulations, it is trickier. Is a surprising result the result of some error, or is it an exciting new discovery? How do we know when to trust a simulation? Throughout this book, we offer some suggestions for developing the sort of intuition and feeling for neuronal behavior that will help you to identify “suspicious” results and their possible causes. As described below, the causes can range from mistakes in the use of the simulation language (programming errors), conceptual errors in the model, inappropriate choices of parameters, and numerical inaccuracies due to the wrong size numerical integration step, to the legitimate but unanticipated behavior of a complex system.

2.4.1 Choice of Numerical Integration Technique

A neural simulation program solves a set of coupled equations like Eq. 2.1 by replacing the differential equation with a difference equation that is solved at discrete time intervals. Typically, smaller time intervals lead to greater accuracy but slower execution time, as more time steps are required for the solution over a given time period. A wide variety of numerical integration techniques has been developed to carry out this procedure with the best compromise between speed and accuracy. These fall into two general categories. So-

called *explicit* methods are the simplest, but can require very small time steps in order to avoid numerical instabilities when there are many small compartments in a model. The *implicit* methods are more complex, but are much more stable (Mascagni 1989).

GENESIS provides a choice between several different numerical integration methods, ranging from the crude explicit *forward Euler* method through highly stable implicit methods. These methods are described in Chapter 20. In general, the best method to use depends on the nature of the model. Often, there is a tradeoff between ease of use and computational efficiency. The default integration method is the *exponential Euler* method, which is optimized for the solution of equations that are of the general form of Eq. 2.1 (MacGregor 1987). This is generally the best choice for cell models having only a few compartments, as used in most network simulations. Chapter 20 discusses the use of a generalized version of the algorithm developed by Hines (1984) for implementing the *backward Euler* and *Crank-Nicholson* implicit methods. These are the fastest of the numerical methods used by GENESIS and are stable and accurate when used with relatively large integration steps. These are used for detailed cell models that contain many compartments. However, they require some additional steps in setting up the simulation and make it harder to interactively modify the simulation. For this reason, one usually develops and refines a simulation using the default method and switches to one of the implicit methods if additional speed is needed for long simulation runs. The *Cable* tutorial which is described in Chapter 5 allows you to experiment with the various integration methods that are available.

2.4.2 Integration Time Step

Even after you have selected a numerical integrator, the step size used in the numerical integration of the differential equations that describe the model is important. The difficulty is that certain combinations of parameters can sometimes result in a situation where the step size is too large to yield accurate results. On the other hand, the use of too small a time step may lead to round off errors, as well as unnecessarily slow execution of the simulation. In general, the step size should be much smaller than the time scale for the most rapidly occurring events. For example, the action potentials that are produced in a simulation typically rise to their maximum value in about 1 *msec*. Thus, a time step of 0.01 *msec* is an appropriate choice.

The default value of the time step that is given for these tutorials usually results in a good compromise between accuracy and speed of computation. However, if you have made significant changes in the default parameters for a simulation, it would be a good idea to experiment with the step size. If increasing it makes no changes in the results, you can use the larger step size to speed up the simulation. If decreasing the step size causes the results to change, you should continue to decrease it until you see no significant changes. Several exercises in the tutorials that follow deal with this question.

2.4.3 Accuracy of GENESIS

Finally, even if a modeler has been extremely careful in constructing a particular model, its accuracy is still dependent on the software in which it is coded. This is especially the case if one is using a general-purpose simulation system such as GENESIS. For this reason, when choosing a neural simulator, the reliability and accuracy of the simulator is at least as important a consideration as its speed. To quantify the speed and accuracy of both GENESIS and other simulators, we have developed the “Rallpacks” suite of benchmarks (Bhalla, Bilitch and Bower 1992). The Rallpacks are currently based on three sets of benchmarks: a linear passive cable with many compartments; a highly branched cable; and a linear axon containing Hodgkin-Huxley channels. In the first two cases, simulator results can be compared to exact analytic solutions. In the third case, which already is an example where the complexity of the model makes analytic solutions impossible, results can be compared to other frequently used, but independently developed simulators. These measures demonstrate that GENESIS is as fast and accurate as any existing simulation system.¹ However, as models within GENESIS or any simulation system become more complex, modelers must be more and more skeptical, vigilant and self-critical.

¹The set of Rallpack benchmarks may be obtained by *ftp* in the same manner as GENESIS, following the procedure outlined in Appendix A.

