RICE UNIVERSITY

Model Reduction of Large Spiking Neurons

by

Anthony Richard Kellems

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

THESIS COMMITTEE:

Steven J. Cox, Advisor
Professor of Computational and Applied Mathematics

Danny C. Sorensen
Noah G. Harding Professor of Computational and Applied Mathematics

Athanasios C. Antoulas
Professor of Electrical and Computer Engineering

Mark Embree

Mark P. Embree
Professor of Computational and Applied Mathematics

Harel Z. Shouval
Assistant Professor of Neurobiology and Anatomy
University of Texas Health Science Center at Houston

HOUSTON, TEXAS

MAY 2010

Abstract

Model Reduction of Large Spiking Neurons

by

Anthony Richard Kellems

This thesis introduces and applies model reduction techniques to problems associated with simulation of realistic single neurons. Neurons have complicated dendritic structures and spatially-distributed ionic kinetics that give rise to highly nonlinear dynamics. However, existing model reduction methods compromise the geometry, and thus sacrifice the original input-output relationship. I demonstrate that linear and nonlinear model reduction techniques yield systems that capture the salient dynamics of morphologically accurate neuronal models and preserve the input-output maps while using significantly fewer variables than the full systems. Two main dynamic regimes characterize the voltage response of a neuron, and I demonstrate that different model reduction techniques are well-suited to each regime.

Small perturbations from the neuron's rest state fall into the subthreshold regime, which can be accurately described by a linear system. By applying Balanced Truncation (BT), a model reduction technique for general linear systems, I recover subthreshold voltage dynamics, and I provide an efficient Iterative Rational Krylov Algorithm (IRKA), which makes large problems of interest tractable. However, these approximations are not valid once the input to the neuron is sufficient to drive the voltage into the spiking regime, which is characterized by highly nonlinear behavior. To re-

produce spiking dynamics, I use a proper orthogonal decomposition (POD) to reduce the number of state variables and a discrete empirical interpolation method (DEIM) to reduce the complexity of the nonlinear terms.

The techniques described above are successful, but they inherently assume that the whole neuron is either passive (linear) or active (nonlinear). However, in realistic cells the voltage response at distal locations is nearly linear, while at proximal locations it is very nonlinear. With this observation, I fuse the aforementioned models together to create a reduced coupled model in which each reduction technique is used where it is most advantageous, thereby making it possible to more accurately simulate a larger class of cortical neurons.

Acknowledgments

Many people have made this work possible, especially my advisor Dr. Steve Cox, who first introduced me to applied mathematics, and who has been an excellent mentor, and Dr. Mark Embree, who has been the most influential in shaping my career through our interaction both in and out of the classroom. I also thank my committee members Dr. Dan Sorensen, Dr. Athanasios Antoulas, and Dr. Harel Shouval. Each have contributed to different aspects of modeling, algorithms, and applications that I have used in this thesis, and it has been a pleasure to learn from them.

I also thank our department secretary Daria Lawrence for her invaluable help with making sure I did everything on time and in the right way. She also has contributed much candy to my insatiable appetite.

I am grateful as well to our department staff Brenda Aune, Ivy Gonzales, Fran Moshiri, and Jennifer Trevino for their assistance with reimbursements, scheduling, and other departmental matters. They have also donated bags of ice for my numerous minor athletic injuries.

Also, I thank Dr. Jan Hewitt for sharpening my writing and presentation skills in an unrelenting, brutally honest, and ultimately kind way.

Additionally, I would not have made it out sane had it not been for the friendships I've had. To Marco Enriquez and Jay Raol, thanks for a great 5 years together; we made it! To Katie Ward, Ryan Nong, Jing Ma, Tommy Binford, Sean Hardesty, Joe Young, and Toni Tullius, you have all made my time in the CAAM department particularly exciting. To my closest friends Marcus Pest, BJ Johnson, Stephanie Taylor, Antonios Kontsos, Natalia Castro, Jessica Cruz, Gabi Vega, Leonard Chan, Ed Roche, Scott Berger, Fr. Justin Price, and Fr. Juan Torres, thank you for your friendship and support.

My family has also been a source of great support, and my siblings, older and sometimes wiser, have always been there for me. Mike, Beth, Kris, and Steve, you all have taught me to follow my dreams and work hard, but also to take time to enjoy life.

The people who deserve the most thanks are my parents. Mom and Dad, thank you for all your support before and during my years at Rice, for your encouragement and your belief in me. This would not have been possible without you. I love you both more than I can express, and this thesis is dedicated to you. And guess what? After 9 years you can finally say your son is out of college.

Finally, I thank God for all the talents and gifts He has given me and which I hope have gone to good use here.

A.M.D.G.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis I present model reduction techniques that can be used to simplify and accelerate simulations of single neurons and neuronal networks. The fundamental motivation for pursuing such reductions is the bottleneck created by the large-scale nature of the problems of theoretical neuroscience. The ultimate goal of this field is to provide a working model of a realistic brain, but the computational power required to achieve this end exceeds current technology. Thus, I propose techniques that shrink the size of the simulated systems while preserving their fidelity to the originals, thereby allowing current technology to solve problems orders of magnitude larger than at present.

1.1 Motivation

The brain is the most complicated organ in animals, containing on the order of 10^{10} neurons (of which there are many different types) which are interconnected via approximately 10^{14} synapses (Shepherd and Koch, 1998). From this extraordinarily complex and heterogeneous network an order emerges whose product is a range of functions, from basic survival responses to higher-level cognitive functions such as memory and emotion (Kandel et al., 2000). However, changes to this order, whether they arise from trauma or from disease, can have life-changing consequences ranging from tremors (as with Parkinson's disease) (Kandel et al., 2000), sudden seizures (as with epilepsy) (Traub and Miles, 1991), or the inability to form new memories (Johnston and Amaral, 1998). In order to diagnose and treat such disorders, it is necessary to understand how the brain works from the small scale of single neurons to the large scale of functional regions.

For example, the hippocampus is a cortical structure that is of critical importance in the formation and storage of memories (Traub and Miles, 1991) as well as in spatial navigation tasks (Witter and Moser, 2006) (Yoganarasimha et al., 2006). Multiple hippocampal regions, each with their own distinct cell types, are involved in these cognitive functions (Ahmed and Mehta, 2009). In order to realize a multi-scale model, one must know the density of connections between these regions, the prototypical morphology and kinetics of each cell type, and the spatio-temporal structure of the inputs to each region. Experimental studies have provided insight into each of these features, thereby enhancing the biophysical accuracy of the resulting models (Koch, 1999). However, having proper initial data is only part of the story; actual simulations allow investigators to validate experimental data and explore new questions. To achieve this, the tools of computational neuroscience must be able to deal with problems on the massive scales described above, which has proven to be a challenge.

1.2 Neuronal Modeling Methods

Currently it is impossible to simulate a biophysically accurate brain, though significant progress has been made toward this goal. The common approach to this problem is to build networks of individual cells and then perform simulations to investigate neural functions. The Blue Brain Project leads this area of research, having reached a milestone of simulating a full cortical column (Markram, 2006). However, these successes generally have come from advances in hardware rather than in modeling methodologies. For example, the Blue Brain Project utilizes thousands of processors running in parallel to achieve its goals, but progression to larger or more detailed simulations will require waiting for Moore's Law to yield corresponding increases in processing power (Lansner, 2009). In contrast, a better model of individual cells will work with existing hardware and, if it is more accurate or faster than current models, will permit the next step in brain simulation to be achieved.

This type of "bottom-up" approach to brain modeling has as its foundation the type of single-cell model that is used. Over a century of research has produced a myriad of model neurons whose complexity has grown steadily. At the simplest level are isopotential cells, which compress the entire neuronal morphology into a single compartment. These can be extremely basic, such as the binary summing units of McCulloch–Pitts models (McCulloch and Pitts, 1943), or they can be more complex and include nonlinear biophysical mechanisms like ion channel kinetics, as in the case of the Hodgkin–Huxley model (Hodgkin and Huxley, 1952). Between these two lies the integrate-and-fire model (Abbott, 1999), which is very common in the literature because of its ease of implementation while still retaining some biophysically meaningful parameters.

Researchers knew, however, that neurons did not behave as isopotential units and therefore incorporated cell morphology into their models. Initially, single dendrites were considered, with Wilfrid Rall being the first to bring a rigorous mathematical description of cable theory to neuroscience (Rall, 1959). The basic form of the cable equation is the same as the heat equation, and hence it is possible to derive analytical solutions in the absence of active ionic mechanisms, as shown by (Rall, 1959) and by Wilson (Wilson, 1999). But since such ionic mechanisms gave rise to the nonlinear spiking behavior that was known to be the method of communication between cells, modeling pursuits shifted from analytical to numerical methods. Rall pioneered this change when he introduced compartmental modeling to the neuroscience community (Rall, 1964).

Multi-compartment modeling soon became standard, allowing both the intricate dendritic structure and the active membrane properties to be incorporated into the models. This spatial extension permitted the inclusion of more detailed biophysics, such as mechanisms describing intracellular processes like calcium exchange, and also allowed observation of neuronal functions such as back-propagating action potentials. Such detail has been facilitated by software suites, such as NEURON (Hines and Carnevale, 2001), which give neuroscientists standardized interfaces through which to build detailed models. With the advent of multi-core workstations and high-performance computing clusters, these tools have been augmented with parallel capabilities (Migliore et al., 2006), permitting ever more realistic simulations. Data acquisition techniques have correspondingly improved along with computing power so that dendritic structure need not be done via manual tracing, a tedious and error-prone process, but rather by automated processes using advanced microscopy techniques and state of the art image processing software (Losavio et al., 2008). Together, the advances in computational and experimental tools have led the field of neuroscience to a point where an almost arbitrary level of detail can be achieved, provided one can wait long enough for the simulation to finish.

1.3 Model Reduction in Computational Neuroscience

In the midst of these rapid increases in model complexity, a fundamental question remains: is such detail necessary to accurately model neuronal behavior? Guided by the maxim of "minimal modeling", researchers continually sought qualitative accuracy in models of much smaller dimension than their full-scale counterparts. Arguably the greatest success was achieved by Traub and Miles (TM) in their 19-compartment reduction of a pyramidal cell (Traub and Miles, 1991), which was even further reduced by Pinsky and Rinzel (PR) to a 2-compartment model (Pinsky and Rinzel, 1994). Though the TM and PR models generate similar outputs to the full-blown pyramidal cell model, they are developed ad hoc (instead of algorithmically) and the inputs and model parameters are not the same as those used in the full model. This disparity in the input-output map makes it difficult for investigators to directly compare the results of experiments to those of simulations that employ the reduced models. A truly reduced model would preserve the input-output relationship of a neuron, but such models have been rare and of limited utility.

Until recently, the only true morphologically accurate model reduction effort has been that of Wilfrid Rall. In 1959 he demonstrated that a passive morphology which satisfies the "3/2" power law and has symmetric synaptic inputs can be collapsed into an "equivalent cylinder", thereby drastically reducing the size of the system to simulate (Rall, 1959). While this was applicable to motoneurons, the assumptions on the morphology and the inputs precluded the use of this technique in general. Development of techniques that are applicable to a broader class of neurons has recently been motivated by a need for efficient computations rather than theoretical derivations.

Nonlinear ion channel kinetics are expensive to evaluate, but sometimes the kinetic variables for different ion channels are similar enough to permit reduction. In 1992, Kepler, Abbott, and Marder introduced the method of "equivalent potentials" to take advantage of this fact (Kepler et al., 1992). By representing the kinetic variables as potentials, they were able to identify dependencies between variables and use these dependencies to reduce the Hodgkin–Huxley model from a four-variable system to a two-variable system, thus permitting faster simulations. This technique was generalized in 2005 by Sorensen and DeWeerth to deal with arbitrary numbers of channel variables as well as to provide an algorithmic, rather than ad hoc, method for computing the reduced system (Sorensen and DeWeerth, 2006). An important drawback of this technique, however, is that it has been developed for single-compartment models; it's applicability to non-uniform (i.e., spatially-varying) kinetics, as would be found in realistic models, has not yet been studied.

Steps toward preserving the nonlinear dynamics for morphologically realistic models have also been motivated by the need for computational efficiency. Though not exactly a model reduction scheme, spatial adaptivity is one way to speed-up simulations without sacrificing nonlinear dynamics. For a CA1 pyramidal cell in the hippocampus, Rempe et al. devised a solution technique in which only areas of the cell which are "active" need to be updated, whereas areas which are quiet need not be (Rempe et al., 2008). They were able to achieve an 80% reduction is computation time for certain problems, but the adaptivity mechanism is dependent upon the synaptic input pattern (Rempe et al., 2008). As a consequence, simultaneous multi-branch stimulation, such as that found in realistic cortical cells, will render this scheme ineffective. While this method did not solve the problem of efficiency in general, it does underscore an important property that is necessary for a truly reduced model: it must be able to handle any general input pattern without severe degradation of computing speed.

Perhaps the closest attempt at incorporating both nonlinear dynamics and morphology in a coherent model reduction framework has come in the past five years. Drawing on tools from computational physics, Woo, Yang and Choi demonstrated that a passive cable can be simulated using an eigenfunction expansion approach using more than an order of magnitude fewer ordinary differential equations (ODEs) and having greater accuracy than the compartmental approach (Woo et al., 2005). Woo and Choi subsequently extended this work to deal with myelinated axons (Woo and Choi, 2007), and, using a pseudo-spectral method, Shin, Yang, and Choi further extended these ideas to work with active cables (Shin et al., 2009).

While in each case these methods produced more accurate results with fewer ODEs, the work raises some questions. First, the authors present no timing comparisons between the full and reduced models, making it hard to determine how computationally efficient this reduction really is. Second, the active cable model used in (Shin et al., 2009) is not a conductance-based model, but a simpler system used by Wilson (Wilson, 1999). Finally, and most importantly, the authors consider input patterns only to single locations along the cable. Although they vary the stimulus location for different examples, they do not give results for simultaneous spatially-distributed inputs, and in fact the dimension of the reduced models appears to depend significantly on the stimulus location. Despite these criticisms, the work from Choi's group demonstrates that nonlinear model reduction is possible in simple cases and bolsters our conviction that drastically reduced models can be found for realistic neuronal models.

1.4 Implications for Network Simulations

The results of this thesis have implications beyond just accurate single cell simulations, and in fact they have profound implications for network simulations. The overarching goal of network reduction is to model the behavior of a network of cells with a system of much smaller dimension. It is not as simple as the single-cell case, however, because often the desired output is an emergent property, meaning that it arises from the interactions of the many individual cells in the network. One such output is the mean firing rate of the network, which can be an important feature when studying brain disorders such as epilepsy. When a network receives too much excitation, or if there is not enough inhibition to counter such excitation, it can fall into a state of seizure, which is marked by extremely high rates of single cell firing, resulting in a highly oscillatory network behavior (Traub and Miles, 1991). Understanding how large networks respond to different input patterns is thus of great importance, but full-scale simulations can be expensive. It makes sense to look for reduced models which capture these salient behaviors using many fewer variables and hence which are faster to simulate.

The network reduction problem has proven to be very difficult and usually has required assumptions which restrict either the type of cells used, the specific inputs, or the synaptic connections. All model reduction efforts thus far have also focused on networks of single-compartment cells. For instance, in 1994 Ermentrout showed that the firing rate of a network of conductance-based cells can be modeled by a simple algebraic formula, but the model requires that synapses be slow (i.e., that they have long time constants) (Ermentrout, 1994). Ermentrout does not suggest that the reduced model will necessarily be a replacement for the full model, but rather that it can be used to find parameter ranges of interest before simulating the full system (Ermentrout, 1994). However, the slow synapse assumption limits the usefulness of this technique.

In 2003 a new technique was proposed by Shriki, Hansel, and Sompolinsky to deal with fast synapses as well (Shriki et al., 2003). The firing rate of a network of conductance-based cells can be characterized using far fewer equations if the f-I curve (the firing rate versus stimulus strength curve) is approximately linear, and if the network is in an asynchronous state (Shriki et al., 2003). There is no assumption on the synaptic time constants, and thus this technique has opened up new brain regions to model reduction, specifically cortical networks, which they studied. Yet, in order to achieve this reduction, the extra assumptions of the f-I curve and the network state prohibit their technique from general use. One can debate that the f-I curve assumption is not too strict, for it is possible that the presence of certain ionic currents can have a linearizing effect (Morel and Levy, 2009). However, restricting the output to be only asynchronous is strong, because it effectively limits the behaviors that can possibly be recovered.

Most recently, Stefanescu and Jirsa have developed a reduction technique based on mode decomposition which gives good qualitative reproductions of network dynamics (Stefanescu and Jirsa, 2008). Although they make certain assumptions on distributions of relevant parameters and the network architecture in order to make the analysis tractable, they are able to recover a broad range of population behaviors with a very small reduced system. While this is an important step, our model reduction goal is to be able to quantitatively reproduce the dynamics of individual cells in the network.

In contrast to these approaches, which deal with simple models of single cells and attempt to reduce the dimensionality of the network itself, the techniques in this thesis can be applied to realistic single cell models. These reduced single cells can then be simulated in a network in order to obtain the full network dynamics without sacrificing properties of the individual neurons or of the input patterns, and thus this thesis provides a new perspective on how to tackle the problem of model reduction of networks.

Chapter 2

Linear Model Reduction

The focus of this chapter is to introduce linear model reduction techniques by considering a basic neuronal model and then extending it naturally to a realistic morphology. Using a single-compartment model, I illustrate the important dynamical features of neurons. This model is then linearized to obtain the quasi-active model in order to illustrate subthreshold behavior. Once these concepts have been introduced in this simple framework, I present the general morphologically accurate model and proceed with linear model reduction. I demonstrate the efficacy of two techniques, one dense and one iterative, and discuss their application to various investigations of behavior in large-scale neurons.

2.1 The Isopotential Cell

One of the most basic isopotential models is the leaky integrate-and-fire (IAF) model, in which the cell can be modeled as a circuit (see Figure 2.1). Following the material found in my Master's thesis (Kellems, 2007), using Kirchhoff's Current Law, the change in voltage v is given by the ordinary differential equation

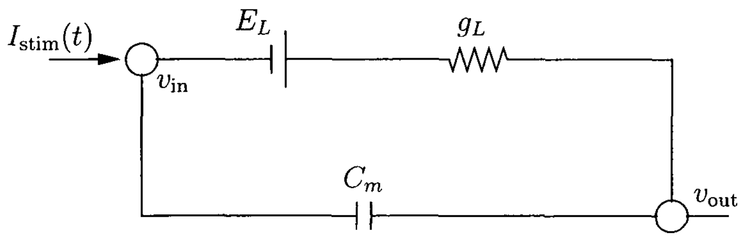


Figure 2.1: Circuit diagram of an integrate-and-fire cell. The voltage v is defined as $v \equiv v_{\text{in}} - v_{\text{out}}$. Slightly edited from my Master's thesis (Kellems, 2007).

```
while $v(t) \leq V_{\text{th}}$
        $C_m v'(t) = -g_L\big(v(t) - E_L\big) + I_{\text{stim}}(t)/A$        (2.1)
end
```

The cell has surface area A and consists of a battery of voltage E_L, a capacitance per unit area C_m, and a resistance per unit area g_L which defines the leakage current due to Cl^- ions. This cell is purely passive, meaning that in the absence of input (i.e., when $I_{\text{stim}}(t) = 0$) the voltage v will decay back to its resting value $\overline{v}$ (in this case $\overline{v} = E_L$). By itself, the ODE in (2.1) does not exhibit spiking behavior, but the **while** condition aims to mimic spiking by instantaneously resetting the voltage to $\overline{v}$ if $v(t) > V_{\text{th}}$, at which time the cell is said to have spiked (or fired).

An experimentally-derived single-compartment model that captures spiking behavior was introduced in 1952 by Hodgkin and Huxley. Through experiments on the squid giant axon, they found that the spiking dynamics could be captured through voltage-gated ionic mechanisms. Specifically, by isolating the Na^+ and K^+ channels, they discovered that the flow of ions could be modeled by gating variables that ac-

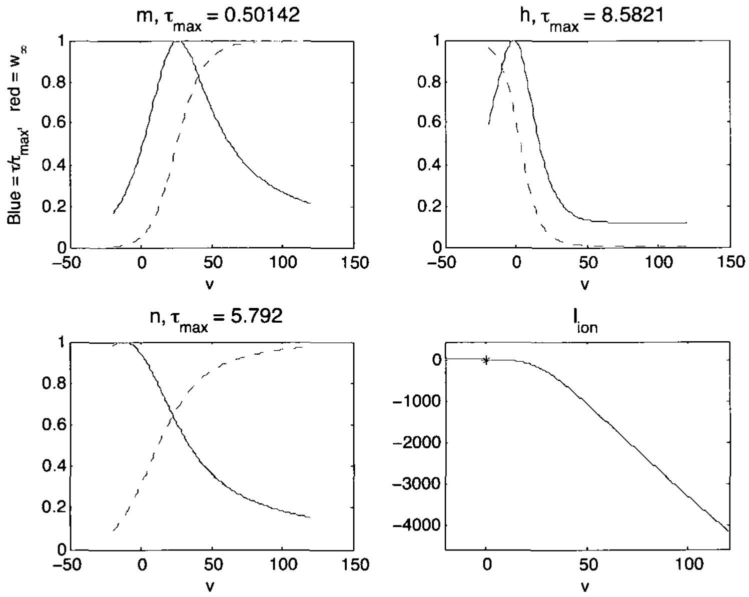


Figure 2.2: Gating variable functions for m, h, and n for the standard Hodgkin–Huxley model.

tivated or inactivated the channels. Each of these gating variables takes on values between 0 and 1, and thus serve to modulate the conductance of their respective ion channels. The ODE for a generic gating variable, call it w, has the form

$$w' = \frac{w_\infty(v) - w}{\tau_w(v)}, \tag{2.2}$$

where w_∞ and τ_w are the voltage-dependent steady-state and time constant functions, respectively. The standard Hodgkin–Huxley model consists of three gating variables, m and h for Na^+ and n for K^+, shown in Figure 2.2.

Now if we write the circuit diagram for this cell as in Figure 2.3, we obtain a capacitor and a battery in parallel with one constant resistor (Cl^-) and two variable

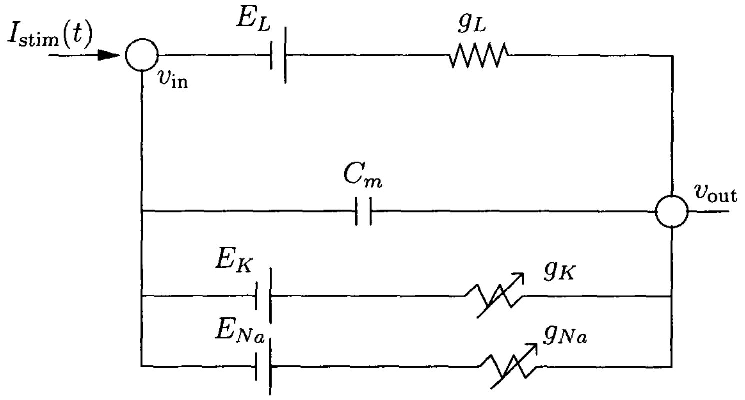


Figure 2.3: Circuit diagram of the Hodgkin–Huxley cell. The voltage v is defined as $v \equiv v_{\text{in}} - v_{\text{out}}$.

resistors (Na^+ and K^+). Applying Kirchhoff's Current Law again yields the following system of ODEs describing the Hodgkin–Huxley (HH) model:

$$
\begin{aligned}
C_m v' &= -g_L(v - E_L) - g_{Na} m^3 h (v - E_{Na}) - g_K n^4 (v - E_K) + I_{\text{stim}}/A && (2.3)\\
m' &= \frac{m_\infty(v) - m}{\tau_m(v)} && (2.4)\\
h' &= \frac{h_\infty(v) - h}{\tau_h(v)} && (2.5)\\
n' &= \frac{n_\infty(v) - n}{\tau_n(v)}. && (2.6)
\end{aligned}
$$

2.2 The Quasi-Active Isopotential Cell

It is well-known that, while linear models do not reproduce spiking dynamics, they do provide information about the frequency response of a system. For neuronal modeling, the linearized systems are good at reproducing the nonlinear system's subthreshold behavior, which is important in certain neuronal functions. Hence it makes sense to consider the linearized version of the HH model in order to address these two concepts.

Noting that $v' = 0$ at rest, this implies $w = \overline{w} \equiv w_\infty(\overline{v})$, and thus $\overline{v}$ is computed by solving the nonlinear equation

$$0 = -g_L(v - E_L) - g_{Na}m_\infty^3 h_\infty(v - E_{Na}) - g_K n_\infty^4(v - E_K), \tag{2.7}$$

for v, which can be done using any standard rootfinding algorithm, such as Newton's Method. With $\overline{v}$ in hand we can proceed with the linearization process.

Following the lead of Koch (Koch, 1999), consider a small stimulus $I_{\text{stim}}(t) = \varepsilon \tilde{I}_{\text{stim}}(t)$. Such a stimulus will give rise to perturbed voltage and gating variables, which are assumed to be of the form

$$\begin{aligned} v &= \overline{v} + \varepsilon\tilde{v} + \mathcal{O}(\varepsilon^2) \\ m &= \overline{m} + \varepsilon\tilde{m} + \mathcal{O}(\varepsilon^2) \\ h &= \overline{h} + \varepsilon\tilde{h} + \mathcal{O}(\varepsilon^2) \\ n &= \overline{n} + \varepsilon\tilde{n} + \mathcal{O}(\varepsilon^2). \end{aligned}$$

Upon substitution into (2.3) we can then solve for the perturbation terms of order ε, which yields

$$C_m\tilde{v}' = -g_L\tilde{\mathrm{v}} - g_{Na}\left[\overline{m}^3\overline{h}\tilde{\mathrm{v}} - (3\overline{m}^2\overline{h}\tilde{m} + \overline{m}^3\tilde{h})E_{Na}\right] - g_K\left[\overline{n}^4\tilde{\mathrm{v}} - 4\overline{n}^3\tilde{n}E_K\right] + \tilde{I}_{\text{stim}}/A \tag{2.8}$$

$$\tilde{m}' = \frac{m_\infty(\overline{v})\tilde{v} - \tilde{m}}{\tau_m(\overline{v})} \tag{2.9}$$

$$\tilde{h}' = \frac{h_\infty(\overline{v})\tilde{v} - \tilde{h}}{\tau_h(\overline{v})} \tag{2.10}$$

$$\tilde{n}' = \frac{n_\infty(\overline{v})\tilde{v} - \tilde{n}}{\tau_n(\overline{v})}. \tag{2.11}$$

This system is called "quasi-active," and since it is linear, it can be written in the standard form

$$z'(t) = Az(t) + Bu(t), \tag{2.12}$$

where

$$z = [\tilde{\mathrm{v}} \quad \tilde{m} \quad \tilde{h} \quad \tilde{n}]^T, \qquad u(t) = \tilde{I}_{\text{stim}}(t), \qquad B = [1 \quad 0 \quad 0 \quad 0]^T,$$

and

$$A = \begin{pmatrix} -1/\tau_L - \overline{m}^3\overline{h}/\tau_{Na} - \overline{n}^4/\tau_K & 3\overline{m}^2\overline{h}E_{Na}/\tau_{Na} & \overline{m}^3E_{Na}/\tau_{Na} & 4\overline{n}^3E_K/\tau_K \\ \overline{m}/\tau_m(\overline{v}) & -1/\tau_m(\overline{v}) & 0 & 0 \\ \overline{h}/\tau_h(\overline{v}) & 0 & -1/\tau_h(\overline{v}) & 0 \\ \overline{n}/\tau_n(\overline{v}) & 0 & 0 & -1/\tau_n(\overline{v}) \end{pmatrix}, \tag{2.13}$$

where the time constants are

$$\tau_L = C_m/g_L, \quad \tau_{Na} = C_m/g_{Na}, \quad \text{and} \quad \tau_K = C_m/g_K.$$

Representing the quasi-active system as a linear system permits both an analytic solution via the eigenvalue decomposition and also a description of the resonant behavior of (2.3) in terms of the eigenvalues of A.

Recall that the linear system given in (2.12) has the analytic solution

$$z(t) = e^{At}\left(\int_0^t e^{-As}Bu(s)ds + z(0)\right). \tag{2.14}$$

Assuming A is diagonalizable, it has an eigenvalue decomposition

$$A = V\Lambda V^{-1}, \tag{2.15}$$

where Λ is a diagonal matrix. That is, the jth column of V is the eigenvector corresponding to the eigenvalue $\lambda_j = \Lambda_{j,j}$. Then substituting (2.15) into (2.14) yields

$$\begin{aligned} z(t) &= Ve^{\Lambda t}\left(\int_0^t e^{-\Lambda s}cu(s)ds + z(0)\right) \\ &= \sum_{j=1}^{n} V_{:,j}e^{\lambda_j t}\left(\int_0^t e^{-\lambda_j s}c_j u(s)ds + z_j(0)\right), \end{aligned}$$

where $c = V^{-1}B$, which is a vector in this example system. The integral can be

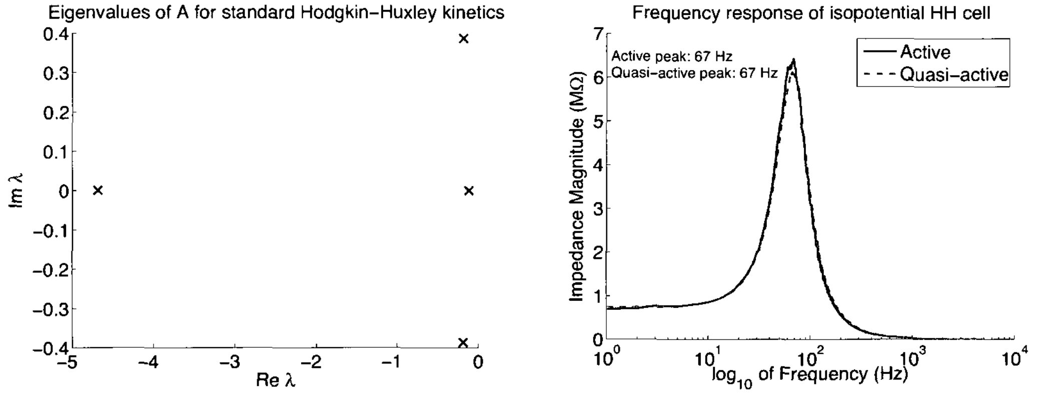


Figure 2.4: Left: Eigenvalues of the 4-by-4 matrix A given in (2.13). Right: Frequency response of the active (solid) and quasi-active (dashed) models using standard HH kinetics.

evaluated numerically via a high-order quadrature scheme such as Clenshaw–Curtis quadrature, as described in (Trefethen, 2007).

Not only does the eigenvalue decomposition yield an analytic solution, but also the eigenvalues of A will tell us whether resonant frequencies exist and, if so, where in the spectrum to look for them. Recall that if λ is an eigenvalue of A and if $\text{Imag}(\lambda) \neq 0$, then the resonant frequency is in the neighborhood of

$$\omega_r = \frac{2\pi \text{Imag}(\lambda)}{1000},$$

where the 1000 in the denominator is to convert from milliseconds to seconds (i.e., from mHz to Hz). If $\text{Real}(\lambda) = 0$ then ω_r is exactly the resonant frequency. However, the quasi-active system is dissipative, meaning that it is possible that some eigenvalues have $\text{Real}(\lambda) \neq 0$, and thus the above equation is only an approximation to ω_r.

Figure 2.4 shows the location in the complex plane of the eigenvalues of the matrix A given in (2.13). There is one complex pair: $\lambda \approx -0.19 \pm 0.38i$. From the imaginary part we obtain $\omega_r \approx 61.5$ Hz. This is close to the actual resonant frequency $\omega_r = 67$ Hz, which can be computed numerically via a parameter sweep or by using a ZAP current

$$I_{\text{ZAP}}(t) = a \sin(bt^c), \tag{2.16}$$

which is a just a sine current with a time-dependent frequency. The frequency response to (2.16) is then computed as the Fourier transform of the voltage divided by the Fourier transform of the input current (Puil et al., 1986).

2.3 Active and Quasi-Active Branched Neurons

The notions of active and quasi-active will now be extended to branched neurons with accurate morphologies, general Hodgkin–Huxley-style kinetics, and spatially-distributed synaptic inputs. These three features of realistic neuronal models make the modeling process more complex by requiring that the following be taken into account:

- Continuity of potential and current balance must be enforced at junction points.
- The "sealed end" condition must be enforced at the distal end of branches with no children.
- Branch radii and ionic conductances can be spatially-varying.

- Any number of ion channels and gating variables can be allowed.
- Either current injection or synaptic conductance may be used as input.

With these assumptions in mind, we introduce and explain the nonlinear cable equation, which is the partial differential equation that will be the focus of the model reduction techniques in this thesis.

The remaining subsections of this chapter come nearly verbatim from my first paper with Roos, Xiao, and Cox (Kellems et al., 2009), which is used here, with some modifications, with kind permission from Springer Science+Business Media: Journal of Computational Neuroscience, Low-dimensional, morphologically accurate models of subthreshold membrane potential, 27(2): 161–176, 2009, A. R. Kellems, D. Roos, N. Xiao, and S. J. Cox, copyright Springer Science+Business Media, LLC 2009.

2.3.1 Nonlinear Cable Equation

We consider dendritic neurons (see, e.g., Figure 2.5) with D branched dendrites meeting at the soma. The dth dendrite possesses B_d branches, and we denote by ℓ_b the length of branch b and encode its radius, as a function of distance from its distal end, by $a_b(x)$. The transmembrane potential along branch b will be denoted by $v_b(x,t)$. We assume that the axial resistivity, R_i (kΩ cm), and membrane capacitance, C_m (μF/cm^2), are uniform throughout the cell. We suppose that branch b carries C distinct currents, with associated densities, $G_{bc}(x)$ (mS/cm^2) and reversal potentials E_c, $c = 1, \ldots, C$. The kinetics of current c on branch b are governed by (powers of)

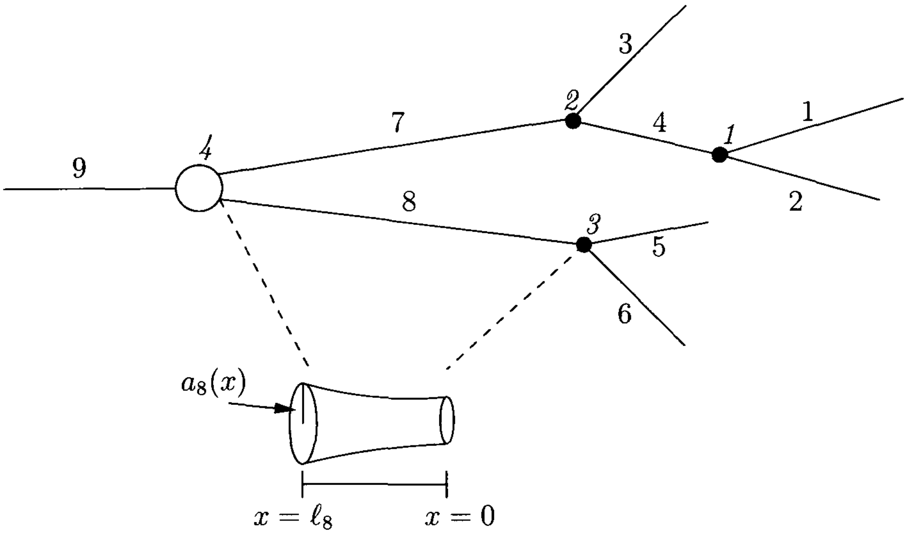


Figure 2.5: Example of a simplified neuron to demonstrate our labeling scheme (taken from my paper with Roos, Xiao, and Cox (Kellems et al., 2009)). Branches are indexed using the Hines ordering (Hines, 1984), and junctions are labeled with italics. For this cell there are $D = 3$ dendrites and $J = 4$ junctions. Our labeling scheme implies, for example, that at junction 3 the children are indexed by $b_3^1 = 5$, $b_3^2 = 6$ and the mother is indexed by $b_3^3 = 8$.

the F_c gating variables, w_{bcf}, $f = 1, \ldots, F_c$. When subjected to input at S_b synapses, these gating variables, together with v_b, obey the nonlinear cable equation

$$\begin{aligned} a_b C_m \partial_t v_b = \frac{1}{2R_i}\partial_x(a_b^2 \partial_x v_b) - a_b \sum_{c=1}^{C} G_{bc}(x)(v_b - E_c) \prod_{f=1}^{F_c} w_{bcf}^{q_{cf}} \\ - \frac{1}{2\pi}\sum_{s=1}^{S_b} g_{bs}(t)\delta(x - x_{bs})(v_b - E_{bs}) \end{aligned} \tag{2.17}$$

$$\partial_t w_{bcf} = \frac{w_{cf,\infty}(v_b) - w_{bcf}}{\tau_{cf}(v_b)}, \qquad 0 < x < \ell_b,\ 0 < t. \tag{2.18}$$

Here g_{bs} (nS) is the time course, x_{bs} is the spatial location, and E_{bs} is the reversal potential of the sth synapse on branch b.

These branch potentials interact at J junction points, where junction J denotes the soma. At junction $j < J$ we denote by b_j^1 and b_j^2 the branch indices of the two children and by b_j^3 the branch index of the mother. Continuity of the potential at each junction requires

$$v_{b_j^1}(\ell_{b_j^1},t) = v_{b_j^2}(\ell_{b_j^2},t) = v_{b_j^3}(0,t), \qquad j = 1,\dots,J-1 \tag{2.19}$$

while current balance there requires

$$a_{b_j^1}^2(\ell_{b_j^1})\partial_x v_{b_j^1}(\ell_{b_j^1},t) + a_{b_j^2}^2(\ell_{b_j^2})\partial_x v_{b_j^2}(\ell_{b_j^2},t) = a_{b_j^3}^2(0)\partial_x v_{b_j^3}(0,t), \qquad j = 1,\dots,J-1. \tag{2.20}$$

The D dendrites join at the soma and associated branch indices are b_J^d, $d = 1,\dots,D$. Continuity of the potential at the soma then reads

$$v_{b_J^1}(\ell_{b_J^1},t) = v_{b_J^d}(\ell_{b_J^d},t), \qquad d = 2,\dots,D. \tag{2.21}$$

We denote this common value by $v_\sigma(t)$ and note that current balance there requires

$$\begin{aligned} C_m\partial_t v_\sigma(t) = {} & \frac{\pi}{A_\sigma R_i}\sum_{d=1}^{D} a_{b_J^d}^2(\ell_{b_J^d})\partial_x v_{b_J^d}(\ell_{b_J^d},t) - \sum_{c=1}^{C} G_{\sigma c}(v_\sigma(t) - E_c)\prod_{f=1}^{F_c} w_{\sigma cf}^{q_{cf}}(t) \\ & - \frac{1}{A_\sigma}\sum_{s=1}^{S_\sigma} g_{\sigma s}(t)(v_\sigma(t) - E_{\sigma s}) \end{aligned} \tag{2.22}$$

$$\partial_t w_{\sigma cf}(t) = \frac{w_{cf,\infty}(v_\sigma(t)) - w_{\sigma cf}(t)}{\tau_{cf}(v_\sigma(t))}, \qquad 0 < x < \ell_b,\ 0 < t, \tag{2.23}$$

where σ is the somatic index and A_σ is the surface area of the soma. Finally, we denote by $\mathcal{L}$ the set of leaf indices, where a leaf is a branch with no children. We suppose that each leaf is sealed at its distal end, i.e.,

$$\partial_x v_b(0,t) = 0, \qquad b \in \mathcal{L}. \tag{2.24}$$

Initially the neuron is at rest, implying that $\partial_t v_b(x,0) = 0$. We solve for the rest state and denote it by $\overline{v}_b(x)$, and similarly for the gating variables, which yields the initial conditions

$$v_b(x,0) = \overline{v}_b(x) \tag{2.25}$$

$$w_{bcf}(x,0) = \overline{w}_{bcf}(x) \equiv w_{cf,\infty}\big(\overline{v}_b(x)\big). \tag{2.26}$$

With the rest state defined, it is easy to modify (2.17) to use current injection instead of synaptic conductance. If we substitute the rest state $\overline{v}_b(x)$ for the $v_b(x)$ in the synaptic input term, then this is equivalent to directly injecting current into the cell, which yields

$$I_{\mathrm{inj},b}(t) \equiv \frac{1}{2\pi}\sum_{s=1}^{S_b} g_{bs}(t)\delta(x - x_{bs})(\overline{v}_b(x) - E_{bs}). \tag{2.27}$$

2.3.2 Linearizing the Cable Equation

The linearization process follows the same technique as that described in §2.2. Consider (2.17) for branch b at its resting potential $\overline{v}_b$. If $g_{bs} = \overline{g}_{bs} + \varepsilon\tilde{g}_{bs}(x,t)$, where

ε is small, then the perturbed voltage and gating variables are assumed to be

$$v_b = \overline{v}_b + \varepsilon \tilde{v}_b + \mathcal{O}(\varepsilon^2) \tag{2.28}$$

$$w_{bcf} = \overline{w}_{bcf} + \varepsilon \tilde{w}_{bcf} + \mathcal{O}(\varepsilon^2). \tag{2.29}$$

Note here that the rest values $\overline{v}_b$ and $\overline{w}_{bcf}$ are now spatially-varying. Substituting (2.28) into (2.17), we construct a linearized model by solving for the perturbation terms $\tilde{v}_b, \tilde{w}_{bcf}$ of order ε. After substitution we find

$$\begin{aligned}\varepsilon a_b C_m \partial_t \tilde{v}_b = \varepsilon \frac{1}{2R_i} \partial_x (a_b^2 \partial_x \tilde{v}_b) - a_b \sum_{c=1}^{C} G_{bc}(x)(\overline{v}_b + \varepsilon \tilde{v}_b - E_c) \prod_{f=1}^{F_c} \left(\overline{w}_{bcf} + \varepsilon \tilde{w}_{bcf}(t) \right)^{q_{cf}} \\ - \frac{1}{2\pi} \sum_{s=1}^{S_b} \left(\overline{g}_{bs} + \varepsilon \tilde{g}_{bs}(t) \right) \delta(x - x_{bs})(\overline{v}_b + \varepsilon \tilde{v}_b - E_{bs})\end{aligned} \tag{2.30}$$

$$\varepsilon \partial_t \tilde{w}_{bcf}(t) = \frac{w_{cf,\infty}(\overline{v}_b + \varepsilon \tilde{v}_b) - (\overline{w}_{bcf} + \varepsilon \tilde{w}_{bcf})}{\tau_{cf}(\overline{v}_b + \varepsilon \tilde{v}_b)}. \tag{2.31}$$

The initial conditions are now

$$\tilde{v}_b(x, 0) = \tilde{w}_{bcf}(x, 0) = 0$$

while boundary conditions, because they are already linear, remain the same as in (2.19), (2.20), (2.24). The soma conditions contain nonlinear terms, but they may be linearized in the same manner as shown here.

Expanding $w_{cf,\infty}$ and τ_{cf} in (2.31) in a Taylor series about $\overline{v}_b$ yields the linearized equations for the gating variables

$$\partial_t \tilde{w}_{bcf}(t) = \frac{w'_{cf,\infty}(\overline{v}_b)\tilde{v}_b - \tilde{w}_{bcf}}{\tau_{cf}(\overline{v}_b)}.$$

In (2.30), if the product of gating variables is written as

$$F_{bc}(\varepsilon) = \prod_{f=1}^{F_c} \left(\overline{w}_{bcf} + \varepsilon \tilde{w}_{bcf}(t) \right)^{q_{cf}}$$

then differentiating with respect to ε yields

$$F'_{bc}(\varepsilon) = \sum_{p=1}^{F_c} \left[q_{cp}(\overline{w}_{bcp} + \varepsilon \tilde{w}_{bcp})^{q_{cp}-1} \prod_{f=1, f\neq p}^{F_c} \left(\overline{w}_{bcf} + \varepsilon \tilde{w}_{bcf}(t) \right)^{q_{cf}} \tilde{w}_{bcp} \right].$$

The order 0 and order ε terms of F_{bc} are

$$F_{bc}(0) = \prod_{f=1}^{F_c} \overline{w}_{bcf}^{q_{cf}} \quad \text{and} \quad F'_{bc}(0) = \sum_{p=1}^{F_c} \left(q_{cp} \overline{w}_{bcp}^{q_{cp}-1} \prod_{f=1, f\neq p}^{F_c} \overline{w}_{bcf}^{q_{cf}} \tilde{w}_{bcp} \right),$$

respectively. Hence the linearized ionic currents take the form

$$I_{\text{ionic}}(x, \tilde{v}_b, \tilde{w}_b) = \sum_{c=1}^{C} G_{bc}(x) \left(F_{bc}(0)\tilde{v}_b + F'_{bc}(0)(\overline{v}_b - E_c) \right).$$

The linearized synaptic input in (2.30) has the form

$$I_{\text{synaptic}}(t, x, \tilde{v}_b) = \frac{1}{2\pi} \sum_{s=1}^{S_b} \left(\tilde{g}_{bs}(t)(\overline{v}_b - E_{bs}) + \overline{g}_{bs}\tilde{v}_b \right) \delta(x - x_{bs}).$$

To complete the linearization process, we equate the terms of order ε and arrive at

$$\partial_t \tilde{v}_b = \mathcal{D}_b \tilde{v}_b + K_b(x)\tilde{v}_b + \sum_{c=1}^{C} \sum_{f=1}^{F_c} d_{bcf}(x)\tilde{w}_{bcf} + \eta_b$$

$$\partial_t \tilde{w}_{bcf}(t) = \phi_{bcf}(x)\tilde{v}_b + \psi_{bcf}(x)\tilde{w}_{bcf},$$

where

$$\begin{aligned}
\mathcal{D}_b &= \frac{1}{2C_m R_i a_b} \partial_x (a_b^2 \partial_x), \\
K_b(x) &= -\frac{1}{C_m} \sum_{c=1}^{C} G_{bc}(x) F_{bc}(0) - \frac{1}{2\pi a_b C_m} \sum_{s=1}^{S_b} \overline{g}_{bs} \delta(x - x_{bs}) \\
d_{bcf}(x) &= -\frac{1}{C_m} G_{bc}(x)(\overline{v}_b - E_c) q_{cf} \overline{w}_{bcf}^{q_{cf}-1} \prod_{p=1, p\neq f}^{F_c} \overline{w}_{bcp}^{q_{cp}} \\
\eta_b &= -\frac{1}{2\pi a_b C_m} \sum_{s=1}^{S_b} \tilde{g}_{bs}(t) \delta(x - x_{bs})(\overline{v}_b - E_{bs}) \\
\phi_{bcf}(x) &= \frac{w'_{cf,\infty}(\overline{v}_b)}{\tau_{cf}(\overline{v}_b)}, \qquad \psi_{bcf}(x) = -\frac{1}{\tau_{cf}(\overline{v}_b)}.
\end{aligned}$$

It is now apparent that this is a linear system for the bth branch, namely

$$\partial_t \begin{pmatrix} \tilde{v}_b \\ \tilde{w}_{b11} \\ \vdots \\ \tilde{w}_{b1F_1} \\ \tilde{w}_{b21} \\ \vdots \\ \tilde{w}_{b2F_2} \\ \vdots \\ \tilde{w}_{bC1} \\ \vdots \\ \tilde{w}_{bCF_C} \end{pmatrix} = \begin{pmatrix} \mathcal{D}_b + K_b & d_{b11} & d_{b12} & \cdots & d_{bCF_C} \\ \phi_{b11} & \psi_{b11} & & & \\ \phi_{b12} & & \psi_{b12} & & \\ \vdots & & & \ddots & \\ \phi_{bCF_C} & & & & \psi_{bCF_C} \end{pmatrix} \begin{pmatrix} \tilde{v}_b \\ \tilde{w}_{b11} \\ \vdots \\ \tilde{w}_{b1F_1} \\ \tilde{w}_{b21} \\ \vdots \\ \tilde{w}_{b2F_2} \\ \vdots \\ \tilde{w}_{bC1} \\ \vdots \\ \tilde{w}_{bCF_C} \end{pmatrix} + \begin{pmatrix} \eta_b \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

or, more concisely,

$$\partial_t z_b = \mathcal{Q}_b z_b + u_b. \tag{2.32}$$

We discretize the neuron in space by dividing each branch into $\gamma_b = \text{ceil}(\ell_b/h)$ compartments, where h is some desired step size. The connectivity of the full morphology is encapsulated in the Hines matrix $\mathbf{H}$, which is the spatial discretization of each $\mathcal{D}_b$ coupled with (2.19) to (2.21) and (2.24) (Hines, 1984). More detail about constructing this matrix will be given in §A.

Using the Hines matrix imposes an outside-in ordering of branches and compartments, which leads to minimal fill-in for Gaussian Elimination (Hines, 1984). If m and n denote, respectively, the number of gating variables per compartment and the total number of compartments, i.e.,

$$m = \sum_{c=1}^{C} F_c \quad \text{and} \quad n = 1 + \sum_{b=1}^{\mathcal{B}} \gamma_b,$$

then the discretized system has $N = n(m+1)$ variables. Restricting our attention to the bth branch, we compartmentalize the voltage by $\tilde{v}_{:,b} \in \mathbb{R}^{\gamma_b}$, and similarly the gating variables, to obtain

$$\partial_t \mathsf{z}_b = \mathsf{Q}_b \mathsf{z}_b + \mathsf{B}_b \mathsf{u}_b$$

where

$$\mathsf{z}_b = \begin{pmatrix} \tilde{v}_{:,b} \\ \tilde{w}_{:,b11} \\ \vdots \\ \tilde{w}_{:,bCF_C} \end{pmatrix} \in \mathbb{R}^{\gamma_b(m+1)}$$

$$\mathsf{Q}_b = \begin{pmatrix} \mathrm{diag}(K_{:,b}) & d_{:,b11} & d_{:,b12} & \cdots & d_{:,bCF_C} \\ \phi_{:,b11} & \psi_{:,b11} & & & \\ \phi_{:,b12} & & \psi_{:,b12} & & \\ \vdots & & & \ddots & \\ \phi_{:,bCF_C} & & & & \psi_{:,bCF_C} \end{pmatrix} \in \mathbb{R}^{\gamma_b(m+1)\times\gamma_b(m+1)}$$

$$\mathsf{B}_b = \begin{pmatrix} I \\ 0 \end{pmatrix} \in \mathbb{R}^{\gamma_b(m+1)\times\gamma_b}, \qquad I \in \mathbb{R}^{\gamma_b\times\gamma_b}, \qquad \mathsf{u}_b \in \mathbb{R}^{\gamma_b}.$$

For example, if branch b receives synaptic input at x_{b1} and x_{b2}, then we have,

$$\mathsf{u}_b = -\frac{1}{2\pi a_b h_b C_m} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{b1}(t)\big(\overline{v}_b(x_{b1}) - E_{b1}\big) \\ 0 \\ \vdots \\ 0 \\ g_{b2}(t)\big(\overline{v}_b(x_{b2}) - E_{b2}\big) \\ 0 \end{pmatrix},$$

where $h_b = \ell_b/\gamma_b$ is the step size on branch b. This term arises as $1/h_b$ when the delta function is discretized. Gathering the branch and soma equations together we define

$$\mathbf{Q} = \begin{pmatrix} \mathsf{Q}_1 & & \\ & \ddots & & \\ & & \mathsf{Q}_{\mathcal{B}} & \\ & & & \mathsf{Q}_\sigma \end{pmatrix} \in \mathbb{R}^{N\times N}, \qquad \mathbf{z} = \begin{pmatrix} \mathsf{z}_1 \\ \vdots \\ \mathsf{z}_{\mathcal{B}} \\ \mathsf{z}_\sigma \end{pmatrix} \in \mathbb{R}^N,$$

$$\mathbf{B} = \begin{pmatrix} \mathsf{B}_1 & & \\ & \ddots & & \\ & & \mathsf{B}_{\mathcal{B}} & \\ & & & \mathsf{B}_\sigma \end{pmatrix} \in \mathbb{R}^{N\times n}, \qquad \mathbf{u} = \begin{pmatrix} \mathsf{u}_1 \\ \vdots \\ \mathsf{u}_{\mathcal{B}} \\ \mathsf{u}_\sigma \end{pmatrix} \in \mathbb{R}^n.$$

Now coupling the branch conditions via the Hines matrix by $\mathbf{A} = \mathbf{Q} + \mathbf{H}$, we arrive at the compartmental model

$$\mathbf{z}'(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t). \tag{2.33}$$

We view $\mathbf{u}$ as the input to the neuron. The neuron's output, $\mathbf{y}$, is the voltage at the site of initiation of action potentials, which is expressed as

$$\mathbf{y}(t) = \mathbf{C}\mathbf{z}(t). \tag{2.34}$$

We suppose that the output is the soma potential, formally written as $\mathbf{y}(t) = \mathbf{z}_{N-m}(t)$. This implies $\mathbf{C} \in \mathbb{R}^{1\times N}$ where $\mathbf{C} = 0$ except for the entry $\mathbf{C}_{1,N-m} = 1$. Together (2.33) and (2.34) give the standard form for linear dynamical systems, where $\mathbf{z}$ is called the *state* vector, $\mathbf{u}$ is the *input* vector, and $\mathbf{y}$ is the *observable* vector. Our goal is to reduce the dimension of this system while exactly preserving the inputs and without sacrificing the accuracy of the soma potential.

2.4 Linear Model Reduction Techniques

I have used two different techniques for linear model reduction. One technique is the classical time-domain method known as Balanced Truncation (BT), which enjoys rigorous theoretical grounding and error analysis but which generally requires dense matrix computations. To combat the computational cost associated with this, I use a newer frequency-domain approach called the Iterative Rational Krylov Algorithm (IRKA). This method achieves essentially the same accuracy as BT in practice, but with an increase in speed of orders of magnitude, thus providing an efficient and accurate algorithm for model reduction of quasi-active systems.

2.4.1 Balanced Truncation

To **balance** a linear dynamical system is to transform it into one where the controllability and observability gramians coincide. As a result, and in a rigorous quantitative sense, the states that are difficult to reach are rarely observed. We present the method below. For the early history, and further details and applications, see (Moore, 1981), (Kailath, 1980), (Antoulas and Sorensen, 2001), and (Antoulas, 2005).

The controllability gramian $\mathcal{P}$ and observability gramian $\mathcal{Q}$ are defined via

$$\mathcal{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*t}dt, \qquad \mathcal{Q} = \int_0^\infty e^{\mathbf{A}^*t}\mathbf{C}^*\mathbf{C}e^{\mathbf{A}t}dt,$$

but typically are computed as solutions to the Lyapunov equations

$$\mathbf{A}\mathcal{P} + \mathcal{P}\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = 0, \qquad \mathbf{A}^*\mathcal{Q} + \mathcal{Q}\mathbf{A} + \mathbf{C}^*\mathbf{C} = 0.$$

We gather the Cholesky factors of the gramians,

$$\mathcal{P} = UU^*, \quad \text{and} \quad \mathcal{Q} = LL^*,$$

and compute the singular value decomposition of the mixed product

$$U^*L = Z\Sigma Y^*.$$

Here Σ is a diagonal matrix whose entries are the eigenvalues of $U^*\mathcal{Q}U$, Z is an orthogonal matrix whose columns are the eigenvectors of $U^*\mathcal{Q}U$, and Y is an orthogonal matrix whose columns are the eigenvectors of $L^*\mathcal{P}L$ (for details on the SVD, see (Trefethen and Bau, 1997)). The diagonal elements of Σ, nonnegative and in descending order, are known as the **Hankel Singular Values** (HSVs) of the dynamical system described by (2.33) and (2.34). We now compose

$$T = \Sigma^{-1/2} Y^* L^*, \quad \text{and} \quad T^{-1} = UZ\Sigma^{-1/2},$$

and note that the transformed gramians

$$\widehat{\mathcal{P}} = T\mathcal{P}T^* \quad \text{and} \quad \widehat{\mathcal{Q}} = T^{-*}\mathcal{Q}T^{-1}$$

are balanced and diagonal in the sense that

$$\widehat{\mathcal{P}} = \widehat{\mathcal{Q}} = \Sigma.$$

Moreover, they are the gramians of the transformed state, $\widehat{\mathbf{z}} = T\mathbf{z}$, which itself is governed by the transformed dynamical system

$$\frac{d\widehat{\mathbf{z}}(t)}{dt} = \widehat{\mathbf{A}}\widehat{\mathbf{z}}(t) + \widehat{\mathbf{B}}\mathbf{u}(t), \qquad \mathbf{y}(t) = \widehat{\mathbf{C}}\widehat{\mathbf{z}}(t), \tag{2.35}$$

where $\widehat{\mathbf{A}} = T\mathbf{A}T^{-1}$, $\widehat{\mathbf{B}} = T\mathbf{B}$, $\widehat{\mathbf{C}} = \mathbf{C}T^{-1}$. Based on the decay of the singular values in Σ, we can construct a reduced model by using only the k largest singular values. This corresponds to approximating (2.35) with

$$\frac{d\boldsymbol{\xi}(t)}{dt} = \widehat{\mathbf{A}}_{11}\boldsymbol{\xi}(t) + \widehat{\mathbf{B}}_1\mathbf{u}(t), \qquad \widehat{\mathbf{y}}(t) = \widehat{\mathbf{C}}_1\boldsymbol{\xi}(t), \tag{2.36}$$

where $\widehat{\mathbf{A}}_{11}$ is the initial $k \times k$ submatrix of $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}_1$ consists of the first k rows of $\widehat{\mathbf{B}}$, and $\widehat{\mathbf{C}}_1$ consists of the first k columns of $\widehat{\mathbf{C}}$.

The reduced state vector $\boldsymbol{\xi} \in \mathbb{R}^k$ has no apparent physiological sense (though see (Freund, 2008) for details about the interpretation of such low-dimensional state vectors) until it is fed to the equation for $\widehat{\mathbf{y}}$, but when $k \ll n$ the computational savings from this model reduction is great. Moreover, contrary to the brute force approach of compartment coarsening, we have not sacrificed the rich input structure. In §2.5.1 we demonstrate that the system in (2.36) allows for numerically exact (near machine precision) reproduction of the soma potential computed from the quasi-active system in a fraction of the time.

2.4.2 Iterative Rational Krylov Algorithm

For small systems, Balanced Truncation is a clean, precise, and feasible method, but as the system dimension grows the cost of computing the gramians becomes prohibitive. Since the reduction comes after the gramians have been computed, this requires storage of dense $N \times N$ matrices before the reduction step, meaning that memory is also an issue.

Some alternative methods for large-scale model reduction of linear systems do exist, such as sparse Lyapunov solvers, as extensively studied in (Sabino, 2006). However, in the context of neuronal modeling here, these methods do not perform well because of the large number of possible inputs. More specifically, the eigenvalue decay of the controllability gramian is much slower than that of the observability gramian, which can be understood intuitively by the fact that there is only one output but the reduced system must be able to accurate represent the action of many inputs. In this case, these Lyapunov solvers converge slowly and are not of practical use in this setting.

Krylov methods, on the other hand, construct approximate reduced systems of a given dimension from the beginning. Instead of transforming the original system and then truncating, the Krylov process iteratively projects the dynamics of the original system onto a smaller subspace. This reduces the memory requirements significantly, since only matrix-vector products are used, and in turn drastically speeds up the reduction process. We now describe the main ideas behind this algorithm; for full details see (Gugercin et al., 2008).

Consider the quasi-active system given in (2.33) and (2.34). We construct two matrices, $\mathbf{V}_k$, $\mathbf{W}_k \in \mathbb{R}^{N\times k}$ such that $\mathbf{z}(t) = \mathbf{V}_k\boldsymbol{\xi}(t)$ for some $\boldsymbol{\xi}(t) \in \mathbb{R}^k$ and such that

$$\mathbf{W}_k^T\big(\mathbf{V}_k\boldsymbol{\xi}'(t) - \mathbf{A}\mathbf{V}_k\boldsymbol{\xi}(t) - \mathbf{B}\mathbf{u}(t)\big) = 0 \tag{2.37}$$

and

$$\mathrm{Range}(\mathbf{V}_k) \cap \mathrm{Null}(\mathbf{W}_k^T) = \{0\}. \tag{2.38}$$

From (2.38) we see that $\mathbf{W}_k^T\mathbf{V}_k$ is invertible. Hence we can use (2.37) to construct a reduced model

$$\frac{d\boldsymbol{\xi}(t)}{dt} = \mathbf{A}_k\boldsymbol{\xi}(t) + \mathbf{B}_k\mathbf{u}(t), \qquad \widehat{\mathbf{y}}(t) = \mathbf{C}_k\boldsymbol{\xi}(t), \tag{2.39}$$

where

$$\mathbf{A}_k = (\mathbf{W}_k^T\mathbf{V}_k)^{-1}\mathbf{W}_k^T\mathbf{A}\mathbf{V}_k, \qquad \mathbf{B}_k = (\mathbf{W}_k^T\mathbf{V}_k)^{-1}\mathbf{W}_k^T\mathbf{B}, \qquad \mathbf{C}_k = \mathbf{C}\mathbf{V}_k. \tag{2.40}$$

The reduced-order system is computed by finding $\mathbf{V}_k$ and $\mathbf{W}_k$ so that the L_2-norm of the error between the transfer functions of the original and reduced systems along the imaginary axis is minimized, i.e., we solve the optimization problem

$$\min_{\mathbf{V}_k,\mathbf{W}_k} \int_{-\infty}^{\infty} \left\|\mathbf{C}^T(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} - \mathbf{C}_k^T(i\omega\mathbf{I} - \mathbf{A}_k)^{-1}\mathbf{B}_k\right\|^2 d\omega.$$

One strategy for solving this is to interpolate the full transfer function, to first order, at the negative of each of its poles. Since these poles are not generally known *a priori* and may be hard to compute, we make an initial guess and then iterate until convergence, indicating that we have arrived at the reduced system. This is achieved in a computationally efficient manner via the Iterative Rational Krylov Algorithm (IRKA), whose implementation details are found in (Gugercin et al., 2008).

I give the IRKA algorithm here for the multiple-input multiple-output (MIMO) case, which is exactly what we have in the quasi-active system.

Algorithm 1 : An Iterative Rational Krylov Algorithm (IRKA) for MIMO Systems

```
Input: Matrices $A$, $B$, and $C$; desired reduced system size $r$;
       shift convergence tolerance $\varepsilon_s$
Output: Reduced order matrices $A_r$, $B_r$, $C_r$
Initialize $A_r$, $B_r$, $C_r$ to be random matrices of appropriate reduced dimensions
Compute shifts $\sigma_i \longleftarrow -\lambda_i(A_r)$ for $i = 1 : r$
Construct $V_r$ and $W_r$ so that
    $\mathrm{Ran}(V_r) = \mathrm{span}\{(\sigma_1 I - A)^{-1}B\tilde{b}_1, \dots, (\sigma_r I - A)^{-1}B\tilde{b}_r\}$
    $\mathrm{Ran}(W_r) = \mathrm{span}\{(\sigma_1 I - A^T)^{-1}C^T\tilde{c}_1, \dots, (\sigma_r I - A^T)^{-1}C^T\tilde{c}_r\}$,
where
    $A_r\tilde{x}_i = \tilde{x}_i\tilde{\lambda}_i, \quad \tilde{y}_i^T A_r = \tilde{\lambda}_i\tilde{y}_i^T$ and $\tilde{y}_i^T\tilde{x}_i = 1$,
    $\tilde{b}_i^T = \tilde{y}_i B_r, \quad \tilde{c}_i = C_r\tilde{x}_i$.
while (relative change in $\sigma_i > \varepsilon_s$) do
    Set $A_r = (W_r^T V_r)^{-1}W_r^T A V_r$, $B_r = (W_r^T V_r)^{-1}W_r^T B$, and $C_r = CV_r$
    $\sigma_i \longleftarrow -\lambda_i(A_r)$ for $i = 1, \dots, r$
    Update $V_r$ and $W_r$ so that
        $\mathrm{Ran}(V_r) = \mathrm{span}\{(\sigma_1 I - A)^{-1}B\tilde{b}_1, \dots, (\sigma_r I - A)^{-1}B\tilde{b}_r\}$
        $\mathrm{Ran}(W_r) = \mathrm{span}\{(\sigma_1 I - A^T)^{-1}C^T\tilde{c}_1, \dots, (\sigma_r I - A^T)^{-1}C^T\tilde{c}_r\}$,
    where
        $A_r\tilde{x}_i = \tilde{x}_i\tilde{\lambda}_i, \quad \tilde{y}_i^T A_r = \tilde{\lambda}_i\tilde{y}_i^T$ and $\tilde{y}_i^T\tilde{x}_i = 1$,
        $\tilde{b}_i^T = \tilde{y}_i B_r, \quad \tilde{c}_i = C_r\tilde{x}_i$
    $A_r = (W_r^T V_r)^{-1}W_r^T A V_r$, $B_r = (W_r^T V_r)^{-1}W_r^T B$, and $C_r = CV_r$
end while
```

2.5 Balanced Truncation Model Reduction Results

We have written a MATLAB suite of functions that loads morphology and channel kinetics and distributions, constructs the associated quasi-active system and both its BT and IRKA reductions, and computes and displays the response of the 4 (nonlinear, quasi-active, BT, IRKA) models to random (in space and time) sequences of synaptic input. These codes are available from the authors upon request. All computations were performed on a Sun Ultra 20 computer with a 2.2 GHz AMD Opteron processor.

Morphologies, shown in Figure 2.6, were obtained from the Rice-Baylor archive (Martinez) and from NeuroMorpho.org (http://NeuroMorpho.org) (Ascoli, 2006), and then imported to our software suite, which lets the user visualize and simulate the neuron via graphical user interfaces. Pyramidal and interneuron ion channel models were obtained from the literature and from ModelDB (http://senselab.med.yale.edu/modeldb) (Hines et al., 2004), as detailed in Tables B.1 and B.2 of the Appendix. Unless otherwise indicated, $\overline{g}_{bs} = 0$.

2.5.1 Dimension Reduction Ratio

Consider a forked neuron as shown in Figure 2.6A. Each of the three branches is 200 μm long and is divided into 2 μm-long segments. The root has radius 2 μm while the leaves have radius 1 μm. Ion channels with Hodgkin-Huxley kinetics (see Table B.1) were uniformly distributed. This leads to a quasi-active system of dimension 1204. We computed BT matrices and found that the Hankel singular values decay

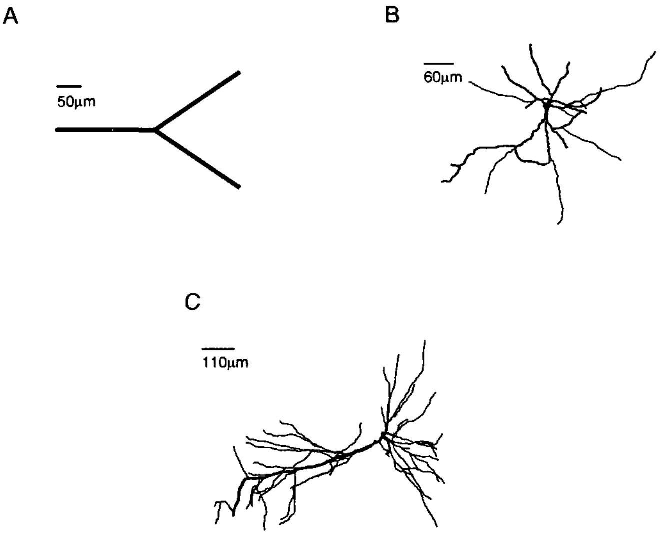


Figure 2.6: Renderings of the three morphologies used in this paper: A) a forked neuron; B) cell AR-1-20-04-A, a projection neuron from rat entorhinal cortex (Martinez); C) cell n408, a rat hippocampal cell from region CA1 (Pyapali et al., 1998).

quite rapidly (Figure 2.7A). To emphasize this, we note that the smallest floating point number ϵ_{mach} such that $1 + \epsilon_{\text{mach}}$ gives a value different than 1 is commonly called *machine precision* and has a value of about 10^{-15}. Indeed, $\sigma_n < \epsilon_{\text{mach}}$ for $n > 65$, indicating that a reduction of at least 20-fold can be obtained. Numerically we observe in Figure 2.7B that, compared to the computed quasi-active soma potential, nearly 5 digits of accuracy can be obtained using only 12 HSVs, a reduction of fully two orders of magnitude. The BT dynamics also track the dynamics of the soma potential from the nonlinear system qualitatively well, though the error is larger (Figure 2.7C).

Tonic synapses have a significant effect on the dynamics, but no effect on the accuracy of the reduced model versus the quasi-active one. We demonstrate the impact of tonic synapses by randomly choosing 10% of the compartments to have

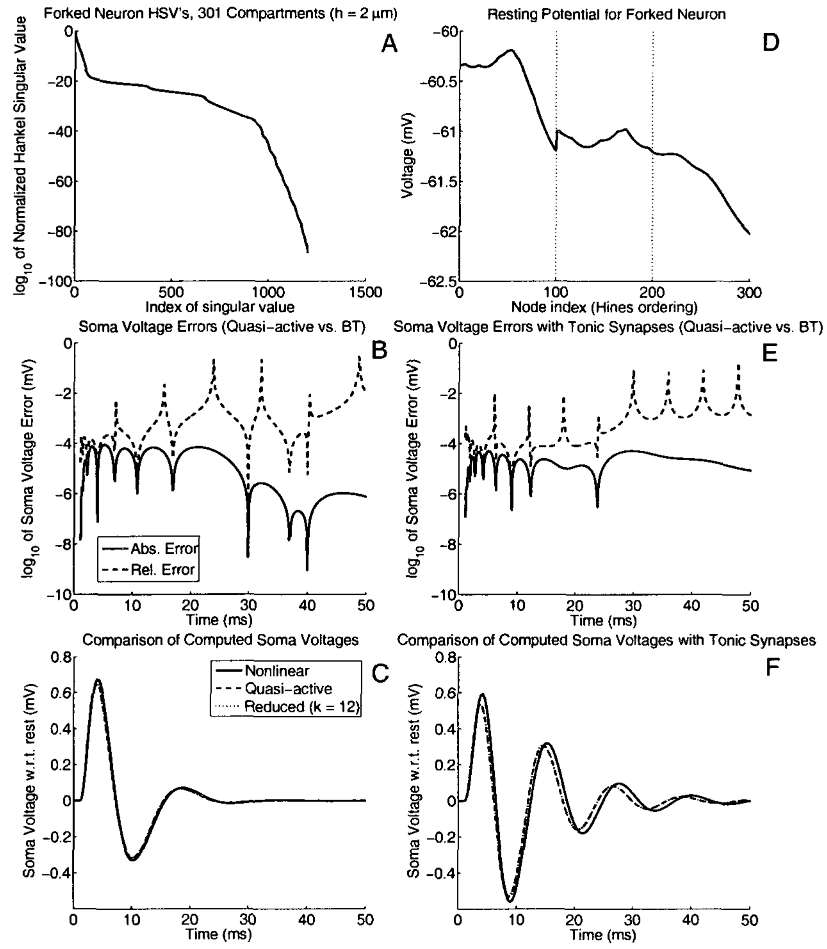


Figure 2.7: A) Hankel singular values for the forked neuron given a 2 μm compartment size. Since there are 301 compartments and the HH ion channels have 3 gating variables, the quasi-active system has 1204 variables; hence the BT system has the same number of HSVs. B) Absolute and relative errors between the computed soma potentials of the quasi-active and BT systems. C) Computed soma potentials from the nonlinear, quasi-active, and BT simulations. At the 1 ms mark a 1 nS alpha-function synaptic input with time constant $\tau = 1$ ms was applied to the midpoint of the top distal branch. Notice that the quasi-active and BT curves are indistinguishable. D) Resting potential of each node for the forked neuron with 10% tonic synapses of $\overline{g}_{bs} = 0.2$ nS each. In the absence of tonic synapses, each node is at rest at ≈ -64.9186 mV. The faint dotted line indicates the end of a branch. E–F) Same as (B–C), but for the case of tonic synapses. Notice that tonic synapses induce more oscillations for the exact same stimulus.

tonic components $\overline{g}_{bs} = 0.2$ nS, which changes the rest potential of the neuron from a uniform value of ≈ -64.9186 mV to one that is now spatially-varying and elevated (Figure 2.7D). Using the exact same stimulus as used for Figure 2.7B–C, we find that the neuron's output is now more oscillatory, but the reduced system's accuracy is unchanged (Figure 2.7E–F).

Computing the BT matrices required about 72 seconds, while the 30 ms simulation required only 0.02 seconds. At first glance this appears expensive when compared to the nonlinear and quasi-active simulation times, both requiring about 1.3 seconds. However, the BT matrix computation is a one-time cost, since these matrices can now be reused to facilitate simulation with their associated morphology.

In fact, it can be seen that the decay of the normalized Hankel singular values almost directly corresponds with the numerical accuracy achieved, as Figure 2.8 illustrates on a more realistic neuron. This result is not surprising, however, given that an error bound exists for BT model reduction (Antoulas and Sorensen, 2001). Therefore, the normalized Hankel singular values may be used as a reliable guide to achieving any desired numerical accuracy compared to the quasi-active system.

2.5.2 Application to Synaptic Scaling

An immediate application for such low-dimensional systems is to accurately quantify how synaptic input scales with distance to the soma, also known as "dendritic democratization" (Magee and Cook, 2000) (Häusser, 2001) (Timofeeva et al., 2008). One standard form of synaptic input is an alpha function, which describes the input

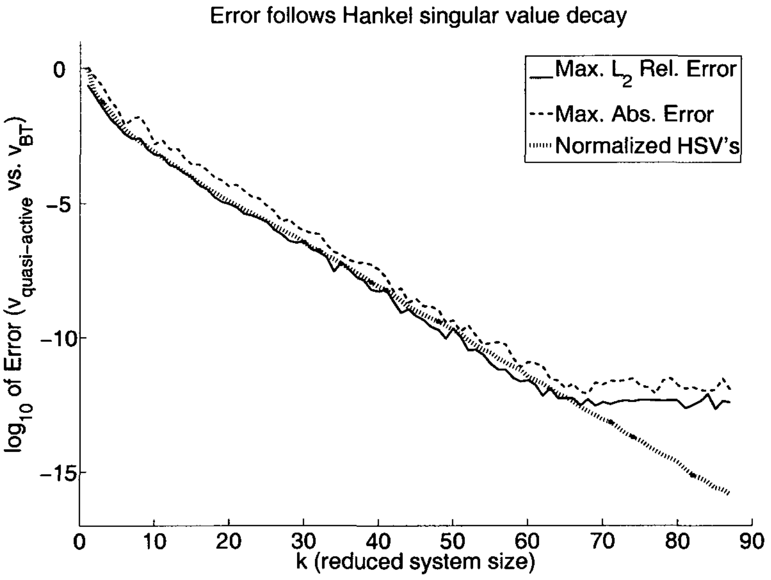


Figure 2.8: Simulations demonstrate converging accuracy for the quasi-active versus reduced system. We used the neuron of Figure 2.6B with the nonuniform channel model of Table B.2. The discretized neuron had 1121 compartments, leading to a 6726-dimensional system. For each reduced system size k, we performed 10 simulations of 50 ms each with 35 random alpha-function synaptic inputs for which $\widehat{g}_{bs} \in [0, 2]$.

onto synapse s of branch b as

$$\tilde{g}_{bs}(t) = \widehat{g}_{bs} \frac{t - t_{\text{start}}}{\tau} \exp\left(1 - \frac{t - t_{\text{start}}}{\tau}\right), \tag{2.41}$$

where t_{start} is the time (in ms) of stimulus onset, and the time constant $\tau = 1$ ms is the time at which the conductance reaches its maximum value of $\widehat{g}_{bs}$. In order to know the strength required for a single synaptic input at a given location to produce a peak target depolarization at the soma, we need to run a parameter sweep for $\widehat{g}_{bs}$. Using the reduced system on the forked neuron with the uniform channels of Table B.1, we execute the bisection method with a tolerance of 10^{-8} to obtain values for

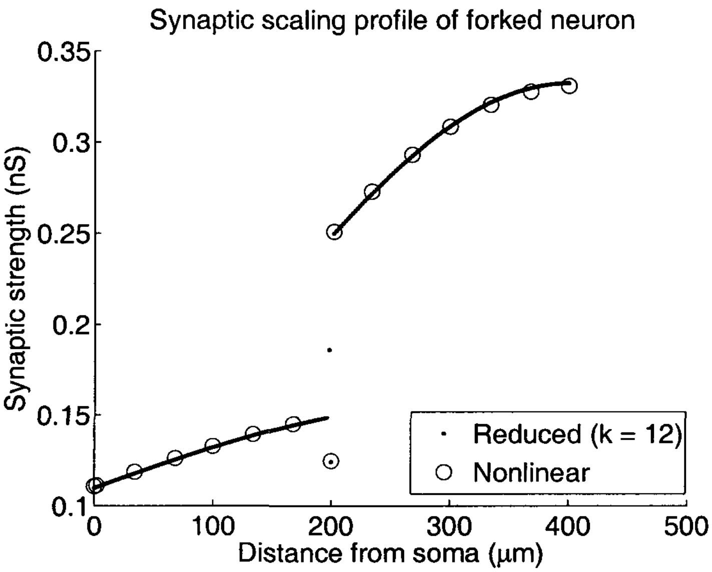


Figure 2.9: Strengths $\widehat{g}_{bs}$ of alpha-function synaptic input needed at each compartment to get a 0.2 mV peak soma depolarization. Dots were computed using the BT system, while circles were computed using the nonlinear system. Notice that the conductance on the root branch increases slowly with increasing distance from the soma, but at the branch point this trend splits. The large difference at the junction occurs because of the disjoint radii of the root branch and the child branches. Only one distinct set of points can be seen from 200-400 μm because the child branches are identical.

each compartment, shown in Figure 2.9, in a total of 4.7 minutes. By contrast, the nonlinear system would require about 100 minutes to do the same.

For more realistic morphologies, and more complex channel models, the one-time cost of computing the BT matrices grows as $\mathcal{O}(n^3)$, but the speed-up gained from using the resulting reduced system is substantial. As an example, neuron AR-1-20-04-A (see Figure 2.6B) was discretized into 1121 compartments, each with length $\approx$ 2 μm. We included I_{Na} and I_K currents following Hodgkin-Huxley kinetics, and used Connor-Stevens kinetics (see Table B.2) (Connor and Stevens, 1971) for the A-type K^+ current I_A whose spatial distribution followed that suggested by (Hoffman et al., 1997). The resulting system had 5 gating variables per compartment, and hence the

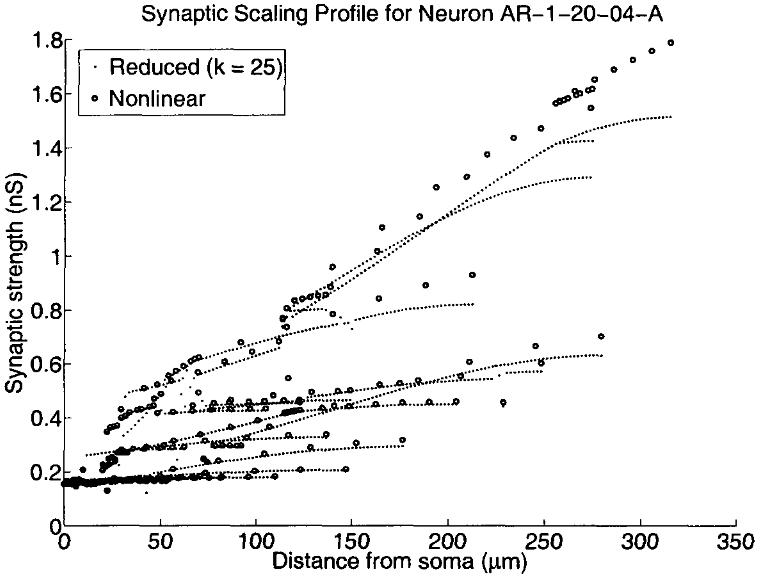


Figure 2.10: Strengths $\widehat{g}_{bs}$ of alpha-function synaptic input needed at each compartment of neuron AR-1-20-04-A to get a 0.2 mV soma depolarization using the BT method. Dots were computed using the reduced system, while circles were computed using the nonlinear system. The reduced system size was $k = 25$.

dimension of the linearized system was 6726. The BT matrices were computed in about 4.6 hours, and setting $k = 25$ we obtained a reduced system that was 269 times smaller than the original linearized system and accurate to more than 5 digits.

Using this reduced model, we executed a parameter sweep for $\widehat{g}_{bs}$ for each compartment, taking a total of 42 minutes. To compare the output to that of the nonlinear model, we did a sweep for 7 compartments in each branch, taking 268 total minutes. The results indicate (see Figure 2.10) that the reduced system accurately tracked the scaling of proximal synaptic inputs while underestimating the scaling for distal conductances. The error in the distal computations is not due to our reduction scheme but rather to our initial quasi-active hypothesis. Regarding Figure 2.10, a value of

$\widehat{g}_{bs}$ for *every* compartment was obtained using the reduced model, whereas to do so for the active model would have required about 21 hours. Hence, to obtain a full portrait of synaptic scaling for this neuron, even factoring in the cost of computing the BT matrices, the reduced system offers a factor of 4 speed-up.

2.5.3 Application to Dendritic Resonance

Neurons have been shown to respond preferentially to stimuli occurring at specific frequencies (Hutcheon and Yarom, 2000) (Hasselmo et al., 2007) (Ulrich, 2002), and quantifying how this resonant frequency varies with stimulus location is a difficult experimental task. However, our reduced model is well-suited to this type of simulation.

Acquiring the resonant frequency data is time-consuming if high frequency resolution is desired. Typical experimental studies stimulate the cell with a ZAP current, which has the form

$$I(t) = a\sin(bt^c + d) \tag{2.42}$$

for some constants a, b, c, and d, with t in ms. The resonant frequency can then be obtained by dividing the Fourier transform of the voltage by the Fourier transform of the input current (Puil et al., 1986). However, this limits the frequency resolution unless very long or detailed simulations are performed. On the other hand, brute force methods give better resolution, but are slow because they require many simulations to be performed using oscillatory currents, such as

$$I(t) = I_0(0.5 + 0.5\sin(2\pi\omega(t - \hat{t})/1000)), \tag{2.43}$$

where I_0 is the peak amplitude, $\hat{t}$ is the time of stimulus onset (in ms), and ω is the frequency (in Hz).

Our reduced model is useful here because we can use the ZAP current to get a band in which the resonant frequency ω_r may be and then run a brute-force search to pinpoint this value, all in much less time than the full system requires. For example, using the ZAP current in (2.42) with parameters $a = 50$, $b = 10^{-6}$, and $c = 3$, we find that the forked neuron with uniform channels has a definite peak near 65 Hz (see Figure 2.11A). Refining this estimate via a brute-force search using (2.43) in a small interval around this peak reveals that ω_r increases linearly as distance from the soma increases.

2.6 IRKA Model Reduction Results

With the BT method as a benchmark, we test the model reduction technique based on using IRKA. First we demonstrate typical convergence of IRKA for our problems, and then we show that cells of much larger dimension can be tackled.

Consider neuron AR-1-20-04-A from Figure 2.6B. We compute the maximum absolute error in the soma voltage between the quasi-active and reduced systems using

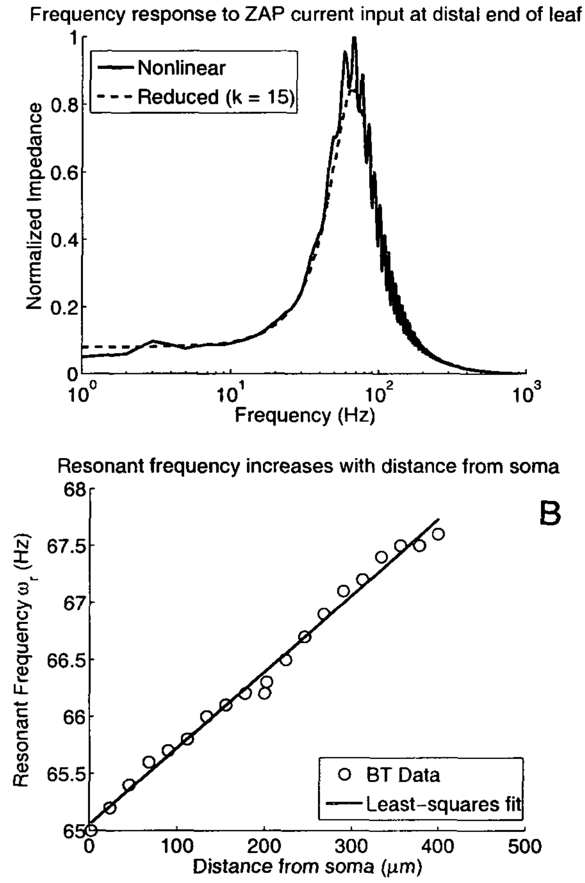


Figure 2.11: A) Frequency-response curves for ZAP current injection at different dendritic locations on the forked neuron with uniform channels. We applied ZAP current for 1000 ms with time-step $\Delta t = 0.1$ ms and computed the normalized impedance versus frequency. Both curves show the same trend, with a peak near 65 Hz, but the reduced system took only 2 seconds, whereas the nonlinear system took 39 seconds. B) Scatter plot of resonant frequencies versus distance from the soma (o), with the least-squares linear fit to the data (thick line). For this experiment, we sampled frequencies every 0.1 Hz on the interval [55, 75] using the current in (2.43) and found that the resonant frequency increases nearly linearly as the stimulus location moves away from the soma.

IRKA (Figure 2.12B), and we compare this to the error curve shown in Figure 2.8A. Notice that nearly 5-digit agreement occurs for a 25-dimensional system using IRKA. This is close to what we would expect from the BT method (20-dimensional system), but IRKA required only 5 seconds to compute the reduced system (Figure 2.12A), whereas BT required 4.6 hours. Thus similar accuracy with IRKA can be obtained

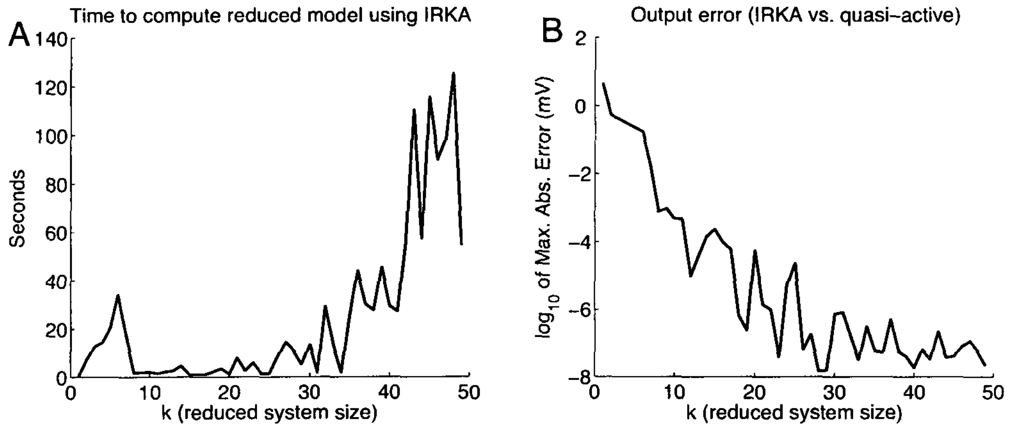


Figure 2.12: A) Time to compute reduced system, and B) maximum absolute error between quasi-active and reduced system output using IRKA for cell AR-1-20-04-A ($N = 6726$).

in a fraction of the time needed for BT. Notice that although the error in IRKA does not decrease as consistently as k increases as it did in BT, nor does the error decrease as rapidly, the accuracy is more than enough for neuroscience applications.

The main difference between IRKA and BT is that IRKA computes a system of a given size, whereas BT computes full matrices that are then truncated to the desired size. If a reduced system of a different size is desired, IRKA must recompute the matrices, whereas BT can just truncate. However, the premium we pay for immediate BT changes is far too expensive, considering that IRKA's results are just as good but are obtained in a fraction of the time. As an example, the reduced system from IRKA with $k = 20$ was used to run the same dendritic democratization parameter sweep as in Figure 2.10. The computed synaptic conductances for the BT and IRKA systems agree to nearly 5 digits for each compartment, indicating that IRKA is indeed computing the right reduced system.

A clear advantage of IRKA is that it handles systems of much larger dimension than BT. We could not use BT for systems where $N > 7000$, due to lack of memory, and even if we could, the computation time would preclude any practical use for such large systems. IRKA does not suffer from this drawback, which translates into the ability to compute reduced models of cells with much finer discretizations and to reduce cells with much more complex dendritic structure.

As an example, we consider neuron n408 (Figure 2.6C), a rat hippocampal cell from region CA1 (Pyapali et al., 1998). We use a 2 μm spatial discretization, yielding 6894 compartments. With a Connor-Stevens ion channel model this gives a total linear system size of $N = 41364$. Using IRKA we find that 5 digits of accuracy can be obtained using a system of dimension $k \approx 15$, which is a 2700-fold reduction, or more than 3 orders of magnitude, in less than a minute (see Figure 2.13). If we use a finer discretization of $h \approx 0.5$ μm, we arrive at a system of size $N = 165330$. IRKA produces a 15-dimensional system that is accurate to nearly 5 digits, a monstrous 11000-fold reduction, or 4 orders of magnitude!

2.7 A Quasi-Integrate-and-Fire Model

Quasi-active neuron models cannot spike, but adopting an integrate-and-fire (IAF) mechanism allows such models to emulate this behavior. Under our assumption that

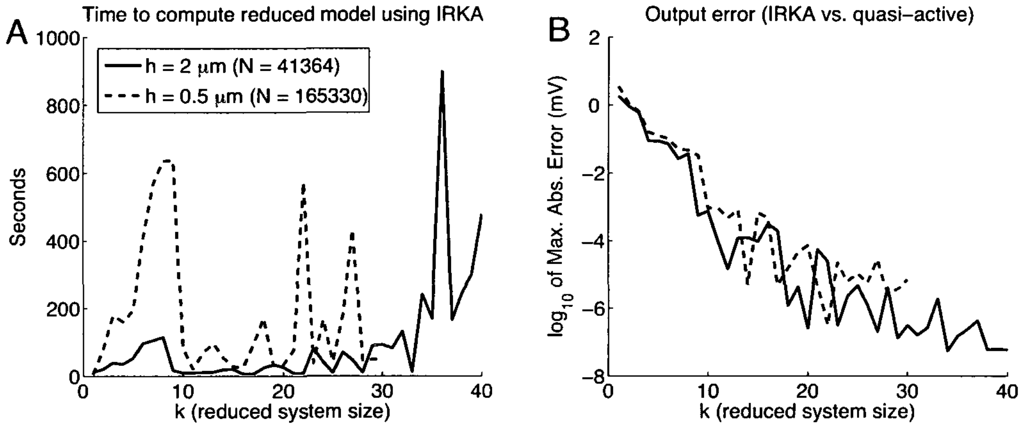


Figure 2.13: A) Time to compute the reduced system, and B) maximum absolute error between the quasi-active and reduced system output using IRKA on cell n408. Solid lines correspond to the $N = 41364$ case, while dashed lines are for the $N = 165330$ case. In (A) the sudden drop in computation time around $k = 10$ occurs for both cases because at that point, though the shifts have not converged, the reduced system is accurate to a tolerance of 3 digits or more

the soma is the site of action potential generation, we only need to check whether

$$\widehat{\mathbf{y}}(t) = \tilde{v}_\sigma(t) \geq V_{\text{th}} \tag{2.44}$$

where V_{th} is some threshold voltage (recall that $\tilde{v}_\sigma(t) \equiv \tilde{v}_1(\ell_{b_j^1}, t))$. At each time-step during the simulation we check if threshold was reached, and if so then we instantaneously reset the soma voltage to the rest and hold it there for a refractory period of τ_{ref} ms.

2.7.1 Thresholding at the Soma

Reset values should be chosen to produce as much similarity as possible between the outputs of the active and quasi-active models while still being biophysically reasonable. However, the state variables of the reduced system do not have apparent biophysical significance until transformed into the observable $\widehat{\mathbf{y}}(t)$. This implies our choice of rest as the reset value for $\boldsymbol{\xi}(t)$.

Using the forked neuron as our test case, we implement the IAF mechanism using the non-uniform channel model. We discretized using $h \approx 1$ μm so that the nonlinear model had 3606 variables; IRKA computed a reduced model having only 10 variables. First we simulate using the nonlinear system, and then we compare this to the output from the reduced system simulation. The spikes generated from both systems are compared to determine if there are any matches. A match is said to occur when the reduced system's spike occurs within τ_{ref} ms before or after a spike in the nonlinear system.

In order to quantify how well the mechanism captures spiking behavior, we compute the coincidence factor $\mathbf{\Gamma}$ (Kistler et al., 1997) (Jolivet et al., 2006), defined as

$$\mathbf{\Gamma} = \frac{N_{\text{coinc}} - N_{\text{nonlin}} N_{\text{reduced}} (\tau_{\text{ref}}/T)}{(N_{\text{nonlin}} + N_{\text{reduced}})(1 - N_{\text{nonlin}}(\tau_{\text{ref}}/T))/2}, \tag{2.45}$$

where N_{nonlin} and N_{reduced} are the number of spikes in the nonlinear and reduced models, respectively, and T is the length of the simulation. The coincidence factor measures how close the spike train from the reduced model approximates that of the nonlinear model by comparing the number of coincident spikes, N_{coinc}, with the number of coincident spikes occurring by chance. $\mathbf{\Gamma}$ is scaled to ensure that $\mathbf{\Gamma} = 1$ implies the spike trains are equal and $\mathbf{\Gamma} = 0$ implies the spike trains would occur purely by chance.

However, $\mathbf{\Gamma}$ can be quite low even when most spikes from the reduced system are coincident. Thus, to better evaluate the effect of false positives, we introduce two metrics: percentage matched, which is the number of spikes from the nonlinear system that are also found in the reduced system; and percentage mismatched, which is how many spikes from the reduced system do not match any spikes from the nonlinear system:

$$\%\text{ matched} = 100\frac{\#\text{ matched}}{\#\text{ nonlinear spikes}} \tag{2.46}$$

$$\%\text{ mismatched} = 100\frac{\#\text{ unmatched reduced spikes}}{\#\text{ reduced spikes}}. \tag{2.47}$$

We vary V_{th} between 8 and 20 mV, with at least 20 simulations at each threshold value, and we use $\tau_{\text{ref}} = 4$ ms throughout. Each simulation lasts 1000 ms, during which time alpha-function synaptic inputs arrive at random locations and at random times.

We ran two sets of simulations to assess the effect of low- and high-activity inputs. The low-activity set used 250 "strong" inputs per simulation, while the high-activity one used 1250 "weak" inputs. These inputs are realistically calibrated by computing $\widehat{g}_{bs}$ values for the reduced system (as in §2.5.2). These values are scaled to obtain synaptic conductances at each compartment that would give approximately 3 mV depolarizations at the soma for the strong inputs, but only 1 mV depolarizations for

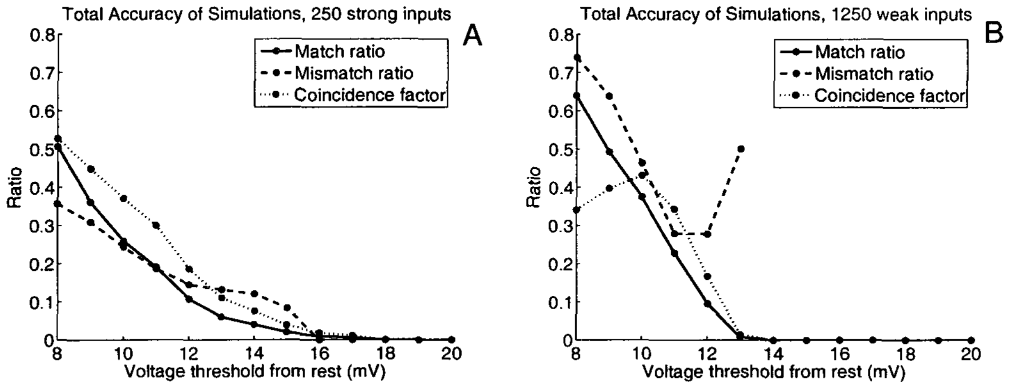


Figure 2.14: Ratio of matched and mismatched spikes, as well as the coincidence factor, as V_{th} increases for A) 250 "strong" inputs and B) 1250 "weak" inputs per simulation. For each V_{th} value we used $\tau_{\text{ref}} = 4$ ms and ran at least 20 simulations of 1000 ms with alpha-function synaptic inputs to random locations. Ratios are given for the match and mismatch values rather than percentages for ease of comparison with Γ.

the weak ones, which agrees with typical values in Chapter 3 of (Traub and Miles, 1991) and (Thomson and Lamy, 2007).

As shown in Figure 2.14, low thresholds capture a large number of the nonlinear spikes, but also yield more mismatched spikes. Fewer mismatched spikes occur as V_{th} increases, but this also leads to greatly diminished numbers of spikes generated, and hence fewer nonlinear spikes captured. More mismatched spikes were observed in the "weak" input simulations, but more spikes were matched as well. The coincidence factors were also low for these simulations, reaching a peak of 0.43 at $V_{\text{th}} = 10$ mV for the "weak" input case and a peak of 0.52 at $V_{\text{th}} = 8$ mV for the "strong" input case.

The spike data is shown explicitly in the soma potentials of Figure 2.15. It is evident from these plots that the reduced system captures the subthreshold behaviors very well even for large depolarizations. Furthermore, the spike times computed by

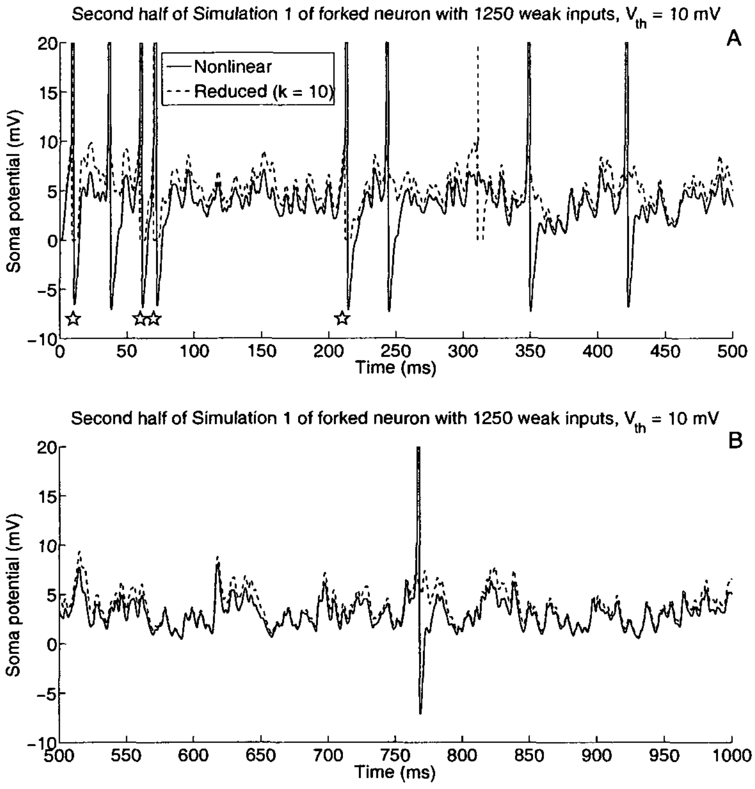


Figure 2.15: A–B) Soma potentials corresponding to the first and second halves of simulation 1 of the "weak" input case using $V_{\text{th}} = 10$ mV. Note that the reduced system spikes are drawn for clarification, and that the voltage axis has been scaled so that subthreshold behavior is more easily seen. Subthreshold behavior is of the nonlinear model is captured well by the reduced model, especially when less firing occurs, and stars below spikes in (A) indicate that four spikes have been captured by the reduced model.

the reduced system tend to be very close to the actual spike times.

2.7.2 Thresholding at Multiple Sites

Of course one has no right to expect high accuracy when applying supra-threshold stimuli to a sub-threshold model. Nonetheless, there are a number of means by which our crude mechanism may be improved. If we fail to detect a somatic spike because it was generated in the dendrite, then it may pay to build the reduction in order that

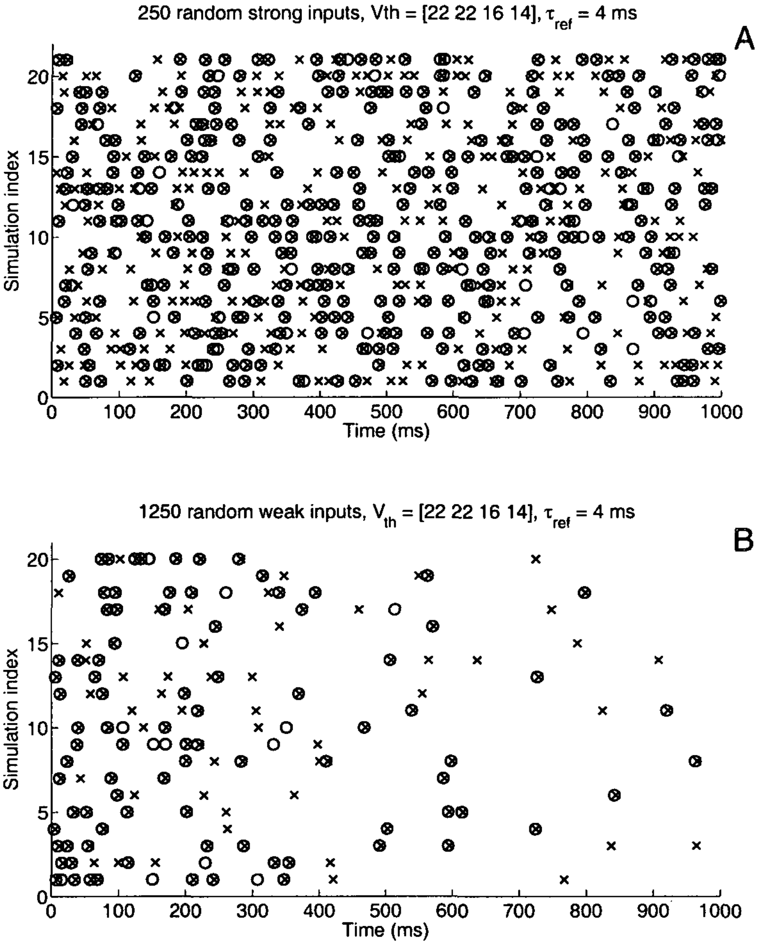


Figure 2.16: Raster diagrams showing spikes computed for the nonlinear system (×'s) and the reduced system (o's) for A) "strong" and B) "weak" input cases demonstrate that a variable-threshold, multi-site IAF mechanism improves accuracy of spike generation. The same input patterns as used for Figure 2.14 were applied to the forked neuron, but this time voltages were observed at the soma and the midpoints of each branch. Threshold values are given in §2.7.2. In total, the "strong" and "weak" input simulations had match scores of 56% and 65%, respectively, and mismatch scores of 15% and 13%, respectively.

it accurately tracks the quasi-active response at several, say p, distinct locations.

Proper investigation of this issue would address the questions of optimal placement of p observers, placement and calibration of threshold mechanisms at each site, and the trade of accuracy for speed as p is increased. We offer here only empirical evidence that such an investigation is indeed warranted.

In particular, we apply a multi-site IAF mechanism to the forked neuron subject to the same input streams that generated Figure 2.14. Since there are 3 branches and a soma, we set $p = 4$ and designate as observables the voltages at the soma and the midpoints of each branch. Threshold values are $V_{\text{th}} = 22$ mV at the leaf midpoints, $V_{\text{th}} = 16$ mV at the root midpoint, and $V_{\text{th}} = 14$ mV at the soma (these values were chosen after manual trial-and-error). Using the same discretization as in §2.7.1, IRKA computes a reduced model with $k = 40$ that is accurate to nearly 5 digits, and hence $\mathbf{A}_k \in \mathbb{R}^{40\times 40}$, $\mathbf{B}_k \in \mathbb{R}^{40\times 601}$, and $\mathbf{C}_k \in \mathbb{R}^{4\times 40}$.

Using the same input patterns as above, we find a significant increase in the accuracy of the spiking behavior (see Figure 2.16) with very little change in computational cost. For the strong input case we match 56% of the actual spikes and have only a 15% mismatch rate. For the weak input case, we get 65% matched with 13% mismatched. The coincidence factor improves in both cases: from 0.43 to 0.73 for the "weak" input case, and from 0.52 to 0.66 for the "strong" input case. Moreover, the difference in simulation time was negligible when compared to the thresholding mechanism of §2.7.1.

Table 2.1: A multi-site variable-threshold IAF mechanism improves spike-capturing accuracy.

Input scheme	N_{nonlin}	N_{reduced}	% Matched	% Mismatched	$\mathbf{\Gamma}$
1250 weak	136	101	64.7	12.9	0.73
250 strong	582	385	56.2	15.1	0.66

Rather than simple thresholding we note that a number of investigations of single compartment models have achieved better spike capturing by incorporating more sophisticated firing mechanisms. For example, (Fourcaud-Trocmé et al., 2003) includes a voltage-dependent exponential term, while (Jolivet et al., 2006) and (Brette and Gerstner, 2005) consider adaptive thresholding. For the class of morphologies and channel distributions of the previous sections, adaptive thresholding, in our hands, did not produce a significant improvement in spike accuracy. Regarding the implementation of an exponential integrate-and-fire model, we note that there is not a natural means by which to nonlinearize our reduced quasi-active system. More precisely, as we observed at the close of §3.1, the reduced state $\boldsymbol{\xi}$ in (2.36) is governed by the small but dense pair $\widehat{\mathbf{A}}_{11}$ and $\widehat{\mathbf{B}}_1$ and hence reflects a complex combination of thousands of gating and voltage variables. The physiological soma potential does not surface until after multiplication by $\widehat{\mathbf{C}}_1$. As such there is not a distinguished voltage term in the differential equation for $\boldsymbol{\xi}$, so it is not clear yet how one might apply a biophysically inspired firing mechanism here.

2.8 Discussion

We have applied two distinct methods to the reduction of dimension of large scale single neuron models. We have demonstrated that in typical settings one may reduce models with tens of thousands of variables to models with merely tens of variables that still accurately track the somatic subthreshold response of spatially distributed synaptic inputs. In regimes where subthreshold stimuli dominate, such as questions of (proximal) dendritic democratization and resonance, our reduced model reveals in minutes what the full models require hours or days to simulate. Although there is no reason to trust our subthreshold integrator on suprathreshold stimuli, we have demonstrated that simply thresholding the reduced voltage at a number of observation points has the potential to capture the cell's full spiking behavior. Further refinement of this approach holds the promise of bringing morphological specificity to the large scale network contributions of (Rudolph and Destexhe, 2006) with significantly fewer compartments than the approach proposed by (Markram, 2006).

Chapter 3

Nonlinear Model Reduction

This chapter, like the previous one, follows almost verbatim from another paper by Kellems, Chaturantabut, Sorensen, and Cox, which has been published in the Journal of Computational Neuroscience (Kellems et al., 2010), which is used here, with some modifications, with kind permission from Springer Science+Business Media: Journal of Computational Neuroscience, Morphologically accurate reduced order modeling of spiking neurons, published online March 19, 2010, A. R. Kellems, S. Chaturantabut, D. C. Sorensen, and S. J. Cox, copyright Springer Science+Business Media, LLC 2010.

Here we extend the results of our previous work to reproduce the full nonlinear behavior of morphologically accurate models by applying two model reduction techniques. The first reduces the number of state variables, while the second reduces the complexity of the nonlinear term by interpolating at a small number of dendritic locations. These techniques preserve the spatial precision of synaptic input while reproducing the global voltage dynamics, including both subthreshold and spiking behaviors.

We start in §3.1 with a description of the full model and the solution methods we employ. In §3.2 we apply the model reduction techniques to simulation data to arrive at the reduced system. Using this framework, in §3.3 we examine the accuracy of the reduced system on simplified morphologies and discuss challenges that branched cells pose. We promptly introduce algorithms to tackle these challenges, and show that they succeed for simple branched cells. In §3.4–3.5 we show that these techniques accurately reproduce the spiking dynamics of a broad class of realistic cells. We end this section with a discussion of applications, improvements, and extensions in §3.6.

3.1 Nonlinear Cable Equation

Using the derivation of the nonlinear cable equation given in §2.3.1 we can proceed with nonlinear model reduction.

It will be useful to consider not only synaptic input but also direct current injection, for reasons which will become apparent in this section. With the rest state defined, it is easy to modify (2.17) to use current injection instead of synaptic conductance. If we substitute the rest state $\overline{v}_b(x)$ for the $v_b(x)$ in the synaptic input term, then this is equivalent to directly injecting current into the cell, which yields

$$I_{\text{inj},b}(x,t) \equiv \frac{1}{2\pi}\sum_{s=1}^{S_b} g_{bs}(t)\delta(x - x_{bs})(\overline{v}_b(x) - E_{bs}). \tag{3.1}$$

If we now partition the cell into N compartments, with C distinct ionic currents per compartment and at most F gating variables per current, then we arrive at the following system of ordinary differential equations

$$\mathtt{v}'(t) = \mathtt{H}\mathtt{v}(t) - \big(\Phi(\mathtt{w}(t))\mathsf{e}\big).\mathtt{v}(t) + \Phi(\mathtt{w}(t))\mathrm{E}_i + \mathtt{G}(t).(\mathtt{v}(t) - \mathrm{E}_s), \quad \mathtt{v}(t) \in \mathbf{R}^N \tag{3.2}$$

$$\mathtt{w}'(t) = \big(\mathtt{A}(\mathtt{v}(t)) - \mathtt{w}(t)\big)./\mathtt{B}(\mathtt{v}(t)), \quad \mathtt{w}(t) \in \mathbf{R}^{N\times C\times F} \tag{3.3}$$

where $\mathtt{H}$ is the N-by-N Hines matrix, $\mathsf{e} = [1\ 1\ \cdots\ 1]^T \in \mathbf{R}^C$,

$$\Phi : \mathbf{R}^{N\times C\times F} \to \mathbf{R}^{N\times C}, \quad \mathtt{G} : \mathbf{R} \to \mathbf{R}^N,$$

$$\mathtt{A} : \mathbf{R}^N \to \mathbf{R}^{N\times C\times F} \quad \text{and} \quad \mathtt{B} : \mathbf{R}^N \to \mathbf{R}^{N\times C\times F},$$

E_i is the vector of channel reversal potentials

$$\mathrm{E}_i = [E_{Cl}\ E_{Na}\ E_K\ \cdots]^T \in \mathbf{R}^C,$$

E_s is the vector of synaptic reversal potentials

$$\mathrm{E}_s = [E_{ex}\ E_{ex}\ E_{in}\ \cdots]^T \in \mathbf{R}^N,$$

and the 'dot' operator, $\mathsf{a}.\mathsf{b}$, denotes element-wise multiplication and $\mathsf{a}./\mathsf{b}$ denotes elementwise division. In this section, we use only excitatory synapses with $E_{ex} = 0$

mV. Here $\mathtt{A}$ and $\mathtt{B}$ are pointwise functions whose actions are defined, recall (2.18), by

$$\mathtt{A}(\mathtt{v})_i = w_{:,\infty}(\mathtt{v}_i), \quad \mathtt{B}(\mathtt{v})_i = \tau_{:}(\mathtt{v}_i), \quad i = 1, \ldots, N. \tag{3.4}$$

We discretize (3.2)–(3.3) following the second-order staggered Euler scheme of (Hines, 1984). More precisely, for a fixed time-step Δt, we evaluate

$$\mathtt{G}^{(j)} = \mathtt{G}((j{-}3/2)\Delta t), \quad \mathtt{w}^{(j)} \approx \mathtt{w}((j{-}3/2)\Delta t) \quad \text{and} \quad \mathtt{v}^{(j)} \approx \mathtt{v}((j{-}1)\Delta t), \quad j = 1, 2, \ldots$$

via the marching scheme: Given $\mathtt{w}^{(j-1)}$ and $\mathtt{v}^{(j-1)}$, evaluate

$$\mathtt{w}^{(j)} = \left[(2\mathtt{B}(\mathtt{v}^{(j-1)}) - \Delta t).\mathtt{w}^{(j-1)} + 2\mathtt{A}(\mathtt{v}^{(j-1)})\Delta t\right]./\left[2\mathtt{B}(v^{(j-1)}) + \Delta t\right] \tag{3.5}$$

and

$$\mathtt{v}^{(j)} = 2\mathtt{v}_{\mathrm{mid}} - \mathtt{v}^{(j-1)} \tag{3.6}$$

where $\mathtt{v}_{\mathrm{mid}}$ is the solution to the linear system

$$(2I/\Delta t - \mathtt{H} + \mathrm{diag}(\Phi(\mathtt{w}^{(j)})\mathtt{e} + \mathtt{G}^{(j)}))\mathtt{v}_{\mathrm{mid}} = 2\mathtt{v}^{(j-1)}/\Delta t + \Phi(\mathtt{w}^{(j)})\mathrm{E}_i + \mathtt{G}^{(j)}.\mathrm{E}_s, \tag{3.7}$$

where the 'diag' operator takes a vector and transforms it into a diagonal matrix. In the case of current injection, $\mathrm{v}_{\mathrm{mid}}$ is the solution to the linear system

$$\big(2I/\Delta t - \mathtt{H} + \mathrm{diag}(\Phi(\mathtt{w}^{(j)})\mathrm{e})\big)\mathrm{v}_{\mathrm{mid}} = 2\mathrm{v}^{(j-1)}/\Delta t + \Phi(\mathtt{w}^{(j)})\mathtt{E}_i - \mathtt{G}^{(j)}.(\overline{\mathrm{v}} - \mathtt{E}_s), \qquad (3.8)$$

where $\overline{\mathrm{v}}$ is the rest potential of the discretized system.

3.2 The Reduced Cable Equation

We apply two model reduction techniques to this ODE system, both of which use the proper orthogonal decomposition (POD). The first technique generates a low-dimensional basis for the state variables, v and $\mathtt{w}$, while the second generates a low-dimensional basis for the nonlinear term.

3.2.1 Proper Orthogonal Decomposition

Given that $\mathrm{v} \in \mathbf{R}^N$, we wish to find a subspace $\mathcal{U} \subset \mathbf{R}^N$ of dimension $k \ll N$ in which the relevant states v are nearly contained. Specifically, given n "snapshots" of the state variables $\mathcal{X} = [\xi(t_1)\ \xi(t_2)\ \cdots\ \xi(t_n)]$, we wish to find an orthonormal basis $\{\phi_i\}_{i=1}^k \subset \mathbf{R}^N$ that solves the minimization problem

$$\min_{\{\phi_i\}_{i=1}^k} \sum_{j=1}^n \left\| \xi(t_j) - \sum_{i=1}^k (\xi(t_j)^T \phi_i)\phi_i \right\|_2^2, \qquad (3.9)$$

i.e., we desire the k-dimensional basis that best fits, in the least squares sense, the snapshot data (Kunisch and Volkwein, 2002). The proper orthogonal decomposition

(POD) provides a solution to this problem via the singular value decomposition (SVD) (Liang et al., 2002).

To obtain the POD basis we first take "snapshots" of the voltage and nonlinear terms at specific (usually equally-spaced) time points during the simulation. For convenience we denote the nonlinear term by

$$\mathtt{N}(\mathtt{v}(t),\mathtt{w}(t)) \equiv \big(\Phi(\mathtt{w}(t))\mathtt{e}\big).\mathtt{v}(t) - \Phi(\mathtt{w}(t))\mathtt{E}_i. \tag{3.10}$$

We save the values of $\mathtt{v}$ and $\mathtt{N}(\mathtt{v},\mathtt{w})$ at times $t_1, t_2, \dots, t_n$, where $t_j = j\Delta t$. The snapshots are stored column-wise in matrices

$$\begin{aligned} \mathtt{V} &= [\mathtt{v}(t_1)\ \mathtt{v}(t_2)\ \cdots\ \mathtt{v}(t_n)] \\ \mathtt{F} &= [\mathtt{N}(\mathtt{v}(t_1),\mathtt{w}(t_1))\ \mathtt{N}(\mathtt{v}(t_2),\mathtt{w}(t_2))\ \cdots\ \mathtt{N}(\mathtt{v}(t_n),\mathtt{w}(t_n))]. \end{aligned}$$

The matrix $\mathtt{V} \in \mathbf{R}^{N\times n}$ will be used to build the POD basis, while $\mathtt{F} \in \mathbf{R}^{N\times n}$ will be used in §3.2.2.

We begin with the SVDs of the snapshot matrices

$$\mathtt{V} = U\Sigma X^T, \quad \mathtt{F} = W\Lambda Y^T,$$

where $U^TU = I$, $X^TX = I$, $W^TW = I$, $Y^TY = I$, and Σ and Λ are diagonal and non-negative. These diagonal elements are ordered in a descending fashion. We choose $k_v \leq n$ and $k_f \leq n$, set

$$\mathtt{U} = U(:,1\colon k_v) \quad \text{and} \quad \mathtt{W} = W(:,1\colon k_f).$$

The columns of these matrices form the bases which are the solutions to (3.9) for the corresponding snapshot sets (Liang et al., 2002).

To complete the POD reduction, we define the reduced voltage variable by

$$\mathtt{v} = \mathtt{U}\tilde{\mathtt{v}} \tag{3.11}$$

and, upon substitution into (3.2), we arrive at the reduced-order system

$$\tilde{\mathtt{v}}' = \mathtt{U}^T\mathtt{H}\mathtt{U}\tilde{\mathtt{v}} - \mathtt{U}^T\mathtt{N}(\mathtt{U}\tilde{\mathtt{v}}, \mathtt{w}) + \mathtt{U}^T\mathtt{G}.(\mathtt{U}\tilde{\mathtt{v}} - \mathtt{E}_s). \tag{3.12}$$

3.2.2 Reduction of the Nonlinear Term via the Discrete Empirical Interpolation Method

While the dimension of (3.12) is now $k_v \ll N$, the nonlinear term still depends on the full dimension N, which indicates that the system has not been truly reduced. For, the reduced voltage $\tilde{\mathtt{v}}$ must be projected up by $\mathtt{U}$ to the full subspace before we can evaluate the nonlinear term, and the result must be projected back down to the reduced subspace by $\mathtt{U}^T$. We apply an empirical interpolation method to find a set of spatial interpolation points $\mathbf{z} = \{z_i\}_{i=1}^{k_f}$ from which the behavior of the full nonlinear

Algorithm 2 : DEIM (Discrete Empirical Interpolation Method)

```
Input: $\{\mathtt{W}_i\}_{i=1}^{k_f} \subset \mathbf{R}^N$ linearly independent
Output: $\mathbf{z} = [z_1, \ldots, z_{k_f}]^T \in \mathbf{R}^{k_f}$
Define $z_1 = \arg\max_j |\mathtt{W}_1(j)|$
$\mathcal{W} = [\mathtt{W}_1]$, $\mathtt{P} = [e_{z_1}]$, $\mathbf{z} = [z_1]$
for $i = 2$ to $k_f$ do
    Solve $(\mathtt{P}^T\mathcal{W})s = \mathtt{P}^T\mathtt{W}_i$ for $s$
    $r \equiv \mathtt{W}_i - \mathcal{W}s$
    Define $z_i = \arg\max_j |r(j)|$
    $\mathcal{W} \leftarrow [\mathcal{W} \quad \mathtt{W}_i]$, $\mathtt{P} \leftarrow [\mathtt{P} \quad e_{z_i}]$, $\mathbf{z} \leftarrow \begin{bmatrix} \mathbf{z} \\ z_i \end{bmatrix}$
end for
```

term can be approximated, thus reducing the complexity of $\mathtt{N}$ to $k_f \ll N$ (Barrault et al., 2004). This method, originally described for use with finite elements, has been extended to our case of finite differences via the discrete empirical interpolation method (DEIM) as given by (Chaturantabut and Sorensen, 2009).

We begin with the basis $\mathtt{W}$ for the snapshot set $\mathtt{F}$ of nonlinear terms and seek a "maximally independent basis set" for $\mathtt{W}$ (Nguyen et al., 2008). The first interpolation point z_1 is the index of the entry of $\mathtt{W}_1$ with the largest magnitude, where $\mathtt{W}_1$ is the first column of $\mathtt{W}$. For $i = 2, \ldots, k_f$ each point z_i is chosen as the index of the entry of the largest magnitude of the residual vector $r \equiv \mathtt{W}_i - \mathcal{W}s$, where $\mathtt{P}$ is the matrix of the $i-1$ coordinate vectors corresponding to the previous interpolation points, $\mathcal{W}$ is the matrix of the previous $i-1$ DEIM basis vectors for $\mathtt{N}$ (that is, $\mathcal{W} = \mathtt{W}(:, 1 : i-1)$), and s is the coefficient vector of components of $\mathcal{W}$ in $\mathtt{W}_i$ relative to the previous $i-1$

interpolation points, as given in Algorithm 2 (Chaturantabut and Sorensen, 2009).

We define the reduced gating variables by

$$\mathtt{w} = \mathtt{P}\tilde{\mathtt{w}}, \tag{3.13}$$

substitute into the nonlinear term, and apply the matrices computed from the DEIM to obtain

$$\mathtt{N}(\mathtt{U}\tilde{\mathtt{v}}, \mathtt{w}) \approx \mathcal{W}(\mathtt{P}^T\mathcal{W})^{-1}\mathtt{P}^T\mathtt{N}(\mathtt{U}\tilde{\mathtt{v}}, \mathtt{P}\tilde{\mathtt{w}}). \tag{3.14}$$

Substituting (3.13)–(3.14) into (3.12) and (3.3), and applying (3.10), we obtain

$$\tilde{\mathtt{v}}' = \tilde{\mathtt{H}}\tilde{\mathtt{v}} - \big(\mathtt{R}\tilde{\Phi}(\tilde{\mathtt{w}})\mathtt{e}\big).(\mathtt{Z}\tilde{\mathtt{v}}) - \mathtt{R}\tilde{\Phi}(\tilde{\mathtt{w}})\mathtt{E}_i + \mathtt{U}^T\mathtt{G}.(\mathtt{U}\tilde{\mathtt{v}} - \mathtt{E}_s) \tag{3.15}$$

$$\tilde{\mathtt{w}}' = \big(\tilde{\mathtt{A}}(\tilde{\mathtt{v}}) - \tilde{\mathtt{w}}\big)./\tilde{\mathtt{B}}(\tilde{\mathtt{v}}), \tag{3.16}$$

where the reduced functions are

$$\tilde{\mathtt{A}}(\tilde{\mathtt{v}}) \equiv \mathtt{P}^T\mathtt{A}(\mathtt{U}\tilde{\mathtt{v}}), \quad \tilde{\mathtt{B}}(\tilde{\mathtt{v}}) \equiv \mathtt{P}^T\mathtt{B}(\mathtt{U}\tilde{\mathtt{v}}), \quad \text{and} \quad \tilde{\Phi}(\tilde{\mathtt{w}}) \equiv \mathtt{P}^T\Phi(\mathtt{P}\tilde{\mathtt{w}}) \tag{3.17}$$

and

$$\begin{aligned}
\mathtt{Z} &\equiv \mathtt{P}^T\mathtt{U} : \mathbf{R}^{k_v} \to \mathbf{R}^{k_f} \\
\tilde{\mathtt{H}} &\equiv \mathtt{U}^T\mathtt{H}\mathtt{U} : \mathbf{R}^{k_v} \to \mathbf{R}^{k_v} \\
\mathtt{R} &\equiv \mathtt{U}^T\mathcal{W}(\mathtt{P}^T\mathcal{W})^{-1} : \mathbf{R}^{k_f} \to \mathbf{R}^{k_v}.
\end{aligned}$$

Since in (3.17) all are pointwise functions, the matrix $\mathtt{P}^T$ just picks off the entries at the interpolation points $\mathbf{z}$, and thus by recalling (3.4) we find

$$\tilde{\mathtt{A}}(\tilde{\mathtt{v}})_i \equiv w_{:,\infty}\big((\mathtt{Z}\tilde{\mathtt{v}})_i\big), \quad \tilde{\mathtt{B}}(\tilde{\mathtt{v}})_i \equiv \tau_{:}\big((\mathtt{Z}\tilde{\mathtt{v}})_i\big), \quad i = 1, \dots, k_f$$

and, similarly, $\tilde{\Phi}$ just computes the rows of Φ corresponding to the indices $\mathbf{z}$. Hence the reduced functions are of complexity k_f, as desired.

We solve the reduced system using the same staggered Euler scheme. We denote

$$\mathtt{G}^{(j)} = \mathtt{G}((j{-}3/2)\Delta t), \quad \tilde{\mathtt{w}}^{(j)} \approx \tilde{\mathtt{w}}((j{-}3/2)\Delta t) \quad \text{and} \quad \tilde{\mathtt{v}}^{(j)} \approx \tilde{\mathtt{v}}((j{-}1)\Delta t), \quad j = 1, 2, \dots$$

and use the scheme: Given $\tilde{\mathtt{w}}^{(j-1)}$ and $\tilde{\mathtt{v}}^{(j-1)}$, evaluate

$$\tilde{\mathtt{w}}^{(j)} = \big[\big(2\tilde{\mathtt{B}}(\tilde{\mathtt{v}}^{(j-1)}) - \Delta t\big).\tilde{\mathtt{w}}^{(j-1)} + 2\tilde{\mathtt{A}}(\tilde{\mathtt{v}}^{(j-1)})\Delta t\big]./\big[2\tilde{\mathtt{B}}(\tilde{\mathtt{v}}^{(j-1)}) + \Delta t\big] \tag{3.18}$$

and

$$\tilde{\mathtt{v}}^{(j)} = 2\tilde{\mathtt{v}}_{\mathrm{mid}} - \tilde{\mathtt{v}}^{(j-1)}, \tag{3.19}$$

where, for current injection, $\tilde{\mathtt{v}}_{\mathrm{mid}}$ is the solution to the linear system

$$\mathtt{L}^{(j)}\tilde{\mathtt{v}}_{\mathrm{mid}} = \mathtt{r}^{(j)} - \mathtt{U}^T(\mathtt{G}^{(j)}.(\bar{\mathtt{v}} - \mathtt{E}_s)), \tag{3.20}$$

where

$$\begin{aligned}
\mathtt{L}^{(j)} &\equiv 2I/\Delta t - \tilde{\mathtt{H}} + \mathtt{R}\mathrm{diag}(\tilde{\Phi}(\tilde{\mathtt{w}}^{(j)})\mathtt{e})\mathtt{Z} \\
\mathtt{r}^{(j)} &\equiv 2\tilde{\mathtt{v}}^{(j-1)}/\Delta t + \mathtt{R}\tilde{\Phi}(\tilde{\mathtt{w}}^{(j)})\mathtt{E}_i.
\end{aligned}$$

In the case of synaptic conductance, $\tilde{\mathtt{v}}_{\mathrm{mid}}$ is the solution to the linear system

$$\big(\mathtt{L}^{(j)} + \mathtt{U}^T\mathrm{diag}(\mathtt{G}^{(j)})\mathtt{U}\big)\tilde{\mathtt{v}}_{\mathrm{mid}} = \mathtt{r}^{(j)} + \mathtt{U}^T(\mathtt{G}^{(j)}.\mathtt{E}_s). \tag{3.21}$$

3.2.3 Numerical Solution of the Reduced System

We have written a MATLAB software suite that loads morphology and channel kinetics and distributions, simulates the full system and obtains POD and DEIM snapshots, and then simulates the reduced system and displays the response of these models to the desired inputs. Gating variable evaluation was performed via a look-up table similar to (Hines, 1984). These codes are available from the authors upon request. All computations were performed using MATLAB version 7.4 (R2007a) on a Sun Ultra 20 computer with a 2.2 GHz AMD Opteron processor. Detailed simulation documentation can be found in the Supplementary Material of (Kellems et al., 2010).

Three ion channel models were used in this study. One follows Hodgkin–Huxley kinetics, while the second augments this model by including an A-type K^+ current with a spatially-varying conductance density given by

$$G_A(x) = 4.15(1 + x/100), \tag{3.22}$$

where x is the distance (in μm) from the soma. We refer to these models as HH and HHA, respectively. Parameters for these two models are the same as given in Tables B.1 and B.2. The third model seeks to mimic weakly excitable dendrites, following the kinetics and spatially-varying conductance densities of (Migliore et al., 1999). We refer to this model as MIG, and we give its parameters in Table B.3.

3.2.3.1 Computational Notes

Since the soma is large but assumed to be isopotential, its surface area is often greater than that of any dendritic compartment. A common practice in coding the solution to (3.2) is to divide each ODE by the surface area of the corresponding compartment. While this is mathematically sound, it yields entries in `H` (and potentially other quantities) that can vary by orders of magnitude, resulting in poor accuracy for the reduced system. The solution is to not divide through by surface area, which provides accurate results for the reduced system.

For the purpose of analyzing spiking behavior, we need only consider the voltage traces from one site (namely, the soma). It is easy to obtain this voltage $\mathtt{v}_{\mathrm{soma}}$ from the reduced system via (3.11), but computing the full matrix-vector product at each time step becomes expensive. However, noting that $\mathtt{v}_{\mathrm{soma}} = \mathtt{U}(j_{\mathrm{soma}}, :)\tilde{\mathtt{v}}$, it is clear that performing this inner product is much cheaper. Thus we can accelerate the reduced system by saving voltage data at different temporal resolutions: a fine resolution for the soma, and a coarse one everywhere else. For instance, saving the somatic potential every r_s time-steps and saving all other potentials every r_c time-steps, where $r_s < r_c$, will reduce the computational expense of (3.11) while still giving the detail we desire at the soma. In this chapter, all timings reflect the use of $r_s = 1$ and $r_c = 10$.

3.3 Results on Simplified Morphologies

3.3.1 Straight Fiber

Consider a uniform fiber that is 1 mm long with $N = 1401$ compartments with HH kinetics. We generate a reduced model using 200 snapshots over 10 ms (solving (3.5)–(3.8) with $\Delta t = 0.01$ ms) for both the POD and DEIM bases by applying a suprathreshold step current of 500 pA for 1 ms at the distal end. This choice of stimulus location permits the action potential to traverse the whole fiber, thus providing a sufficiently rich set of snapshots from which to build the reduced system (see Figure 3.1A). Computing the singular values for the POD and DEIM snapshot matrices as in (3.2.1), we find that they decay quite rapidly (see Figure 3.1B), thus it is reasonable to expect that a small reduced system can approximate the full system well.

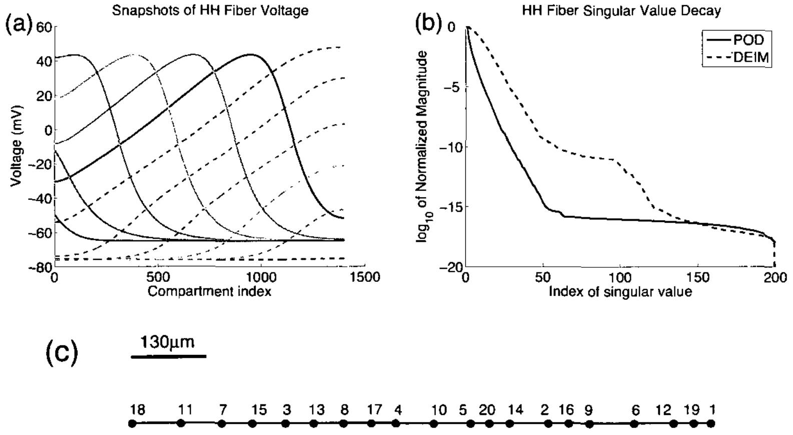


Figure 3.1: (a) Sample traces of voltage snapshots used to construct the HH fiber reduced system. Shown are snapshots 22:10:132 of the simulation. (b) Decay of singular values for the POD and DEIM bases for the HH fiber. (c) Location of the first 20 DEIM points on the fiber, indexed by order in which they were computed.

Note that the decay of these singular values to some tolerance does not necessarily imply a corresponding accuracy of the reduced system as compared to the full one. As shown in §2, for linear systems there exist SVD-based model reduction methods, such as Balanced Truncation, for which rigorous error bounds exist in terms of the decay of Hankel singular values (Glover, 1984) (Antoulas and Sorensen, 2001), and these have been successfully applied to the linearized version of (2.17) (Kellems et al., 2009). For nonlinear systems the reduction relies upon having a good snapshot set. Hence if the snapshot set is poor, then the decay of the singular values need not be indicative of the existence of a good reduced system. However, if the snapshot set is good then the singular value decay can be used as a guide to estimate how large the reduced system must be.

To test accuracy and speed, we perform a series of simulations using random inputs and compare the reduced system's performance with that of the full system. A match is said to occur when a spike, indicated by a somatic depolarization of $V_{\text{th}} = 40$ mV or more from rest, in the reduced system occurs within $\tau_{\text{ref}} = 2$ ms of a spike in the full system. Our metrics for quantifying the spike-capturing accuracy of the reduced system are the same as those in §2.7.1. We use the coincidence factor $\mathbf{\Gamma}$ (Kistler et al., 1997), defined as

$$\mathbf{\Gamma} = \frac{N_{\text{match}} - N_{\text{full}} N_{\text{reduced}} (\tau_{\text{ref}}/T)}{(N_{\text{full}} + N_{\text{reduced}})(1 - N_{\text{full}}(\tau_{\text{ref}}/T))/2}, \tag{3.23}$$

where N_{full} and N_{reduced} are the number of spikes in the full and reduced models, respectively, N_{match} is the number of matches, and T is the length of the simulation. To better evaluate the effect of false positives, we use the percentage matched and percentage mismatched measures of §2.7.1, rewritten here explicitly as

$$\% \text{ matched} = 100 \frac{N_{\text{match}}}{N_{\text{full}}} \tag{3.24}$$

$$\% \text{ mismatched} = 100 \frac{N_{\text{reduced}} - N_{\text{match}}}{N_{\text{reduced}}}. \tag{3.25}$$

We simulate for 1 second with 200 random step currents, each lasting 0–5 ms and having amplitudes of 0–100 pA, applied to random locations on the fiber. To solve both systems we use $\Delta t = 0.1$ ms. We vary the dimension of the reduced system, using $k_v = k_f$, and run 20 simulations at each value (the same 20 input patterns were used for each reduced system). The results, shown in Table 3.1, indicate that reduced systems nearly 100-fold smaller than the original ones reproduce highly accurate spiking patterns and are about 5 times faster.

It should be noted that throughout this thesis we use $k_v = k_f$, which is justified by results of studies we performed on simple morphologies. For these studies we varied k_v and k_f independently and computed the performance metrics for each pair, but we always found that $k_v = k_f$ was the best choice. Hence we use this empirical heuristic even for complex morphologies.

Table 3.1: Performance of reduced model (here $k_v = k_f$) of HH fiber, $N = 1401$, as compared with the full model.

k_v	Speed-up	% Matched	% Mismatched	Γ
10	6.3×	87.4	8.2	0.893
15	5.9×	98.9	1.1	0.988
20	5.6×	99.7	0	0.998
30	4.6×	100	0.6	0.997

To assess the effects of different ion channel models on accuracy and speed, we perform the same experiment, but with a channel model incorporating an A-type K^+ current (HHA model). Due to the decreased excitability of the distal part of the fiber, we increase the number of stimuli to 500 and the range of amplitudes of the step currents to 0–300 pA. As shown in Table 3.2, although the accuracy initially decreases for very small k_v values, it is rapidly regained so that nearly exact spike dynamics ($\mathbf{\Gamma} = 0.990$) are reproduced with $k_v = k_f = 30$, a 46-fold reduction in dimension. Additionally, the reduced systems show greater speed-ups versus the full systems than do the reduced systems using the HH model. Examination of voltage traces (Figure 3.2) shows that not only are the spike times correct, but in fact the sub- and suprathreshold voltage dynamics agree very well.

One may wonder why the speed-up is basically one order of magnitude while the dimension reduction is two orders of magnitude. The reason lies in the fact that the full system matrices are large but sparse, whereas the reduced matrices are small but dense. Hence the associated matrix-vector products have different computational costs, and we ought not expect the same scalings.

Table 3.2: Performance of reduced model (here $k_v = k_f$) of HHA fiber, $N = 1401$, as compared with the full model.

k_v	Speed-up	% Matched	% Mismatched	$\mathbf{\Gamma}$
10	8×	81.7	64.3	0.457
15	7.5×	77.7	7.7	0.840
20	6.9×	95.8	0.5	0.976
30	5.8×	98.6	0.5	0.990

3.3.2 Challenges of Branched Cells

Now consider the forked neuron which has one mother branch and two daughter branches, all with radius 1 μm and length 500 μm. The cell consists of $N = 1501$ compartments having HH kinetics. The first step in constructing a reduced model is to generate a sufficiently rich set of snapshots, but this simple cell illustrates that the task is not trivial.

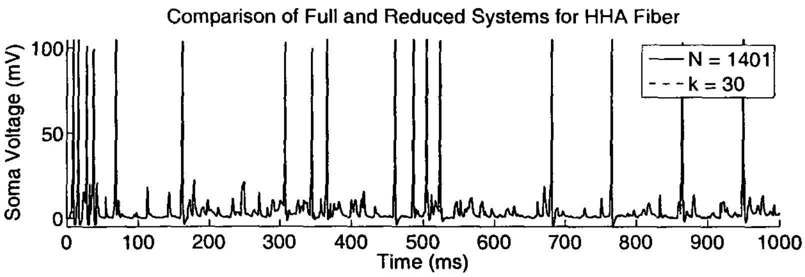


Figure 3.2: Comparison of somatic voltage traces as computed by the full (solid) and reduced (dashed) systems for the HHA fiber. Although two curves are plotted, only one is seen because the reduced system is so accurate ($k_v = k_f = 30$) that it nearly exactly reproduces the dynamics.

3.3.2.1 Phantom Spiking

We desire a reduced model that will reproduce the correct dynamics of the full model independent of the location of the inputs. In order to capture spiking behavior, this implies that the snapshots to be used must contain local descriptions of action potentials. For example, on the fiber we needed to ensure that each node experienced spike dynamics. This was accomplished by initiating a spike at one end and allowing it to travel to the other end (this includes the after-hyperpolarization as well).

For the forked neuron, isolating the local spiking behavior is not so simple because each spike that is initiated will split upon reaching the junction point, and thus at least two branches will contain a spike at the same time. Such a situation is not good, because the snapshots will contain information about both spikes simultaneously, causing *phantom spiking.* This phenomenon consists of stimulus arriving at one location, but, because of the simultaneous spikes in the snapshots, a similar output is observed at a different location on another branch, thus corrupting the computed solution.

3.3.2.2 Branch-Ortho: Branchwise Orthogonalization

To excise these phantom spikes, we simulate the full model with a suprathreshold stimulus applied at a location that will allow a spike to propagate throughout the whole cell; this location should be the soma, a junction point, or the distal end of a leaf in order to prevent two spikes from occurring on the same branch. Let us denote the resulting set of snapshots as $\mathcal{S}$. To isolate the dynamics of branch j we create a local set $\mathcal{L}_j$ which is initially the same as $\mathcal{S}$. The snapshots in $\mathcal{L}_j$ are then modified by setting to rest all the values of elements that do not belong to branch j. The final set of snapshots $\widehat{\mathcal{S}}$ is then the union of the local sets. Hence the snapshots are "orthogonal" in the sense that only one branch is active in each snapshot. While intuitive, this technique is a bit naïve because it completely isolates branches, effectively throwing away information about the coupling that occurs at junctions.

An improvement of this method can partially recover this coupling information by not just generating snapshots of isolated branches, but of connected branches that form continuous routes throughout the dendritic tree. A *route* R is defined as a set of branches in which

- at most one branch is present at each depth in the dendritic tree
- the branch at depth $j-1$ is the parent of the branch at depth j.

Hence the dimension of a route, $\dim(\mathsf{R})$, is equal to the number of branches it contains,

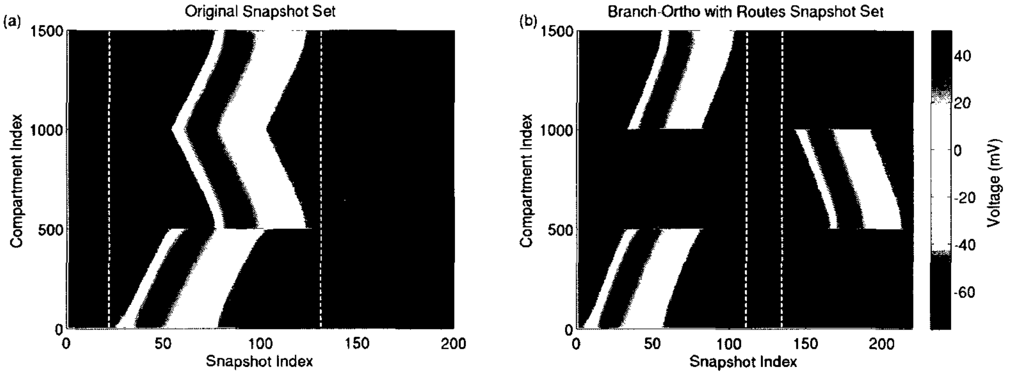


Figure 3.3: Illustration of Branch-Ortho with routes on a forked neuron. (a) The original set of snapshots, with white dashed lines bounding the global active zone obtained via the first application of V-Slim. (b) The resulting snapshot set after the Branch-Ortho algorithm is run. Two routes are used: one consisting of branches 1 and 3, and one of just branch 2. Notice that information about the coupling at the junction is preserved in the first route. White dashed lines in this plot enclose the snapshots to be removed, as computed via the second V-Slim application.

and the length of a route is the sum of the lengths of the branches it contains, i.e. $\text{len}(\mathsf{R}) = \sum_{b \in \mathsf{R}} \ell_b$.

Routes are useful because if we are given a set of snapshots in which an action potential travels from the soma throughout the dendritic tree, then routes form long, continuous snapshots of the dynamics and maintain coupling information between parents and children (see Figure 3.3). Not only does using routes improve accuracy, but it further shrinks the size of the snapshot set. Since the active zones of the parent and child branches overlap, routes yield more information in fewer snapshots than the naïve method.

Which routes should comprise the Branch-Ortho snapshot set? Let $\mathcal{R}$ denote a set of routes and let $\dim(\mathcal{R})$ be defined as the number of routes in the set. We define $\mathcal{R}$ to be *optimal* if

- it contains every branch of the cell
- all routes are non-intersecting (i.e., no branch appears in more than one route)
- it contains the fewest routes possible,

and we denote an optimal set of routes by $\widehat{\mathcal{R}}$. After a little thought, we can conclude that *we can build an optimal set of routes entirely of routes which begin with leaves.* Thus we know that if there are L leaves in a cell, then we need to store only L routes, and hence $\dim(\widehat{\mathcal{R}}) = L$. Thus the smaller the ratio of $L/\mathcal{B}$ the better this method should be. It also follows that every $\widehat{\mathcal{R}}$ contains at least one R such that $\dim(\mathsf{R}) = 1$. Algorithm 3 implements Branch-Ortho with a set of optimal routes, which is obtained by proceeding outside-in from leaves to soma.

3.3.2.3 V-Slim: A Snapshot Elimination Algorithm

In order to implement the Branch-Ortho algorithm in any generality, we must ensure that the snapshot set does not become so large that computing the SVDs of the POD and DEIM inner product matrices becomes prohibitively slow. Since the SVD scales as $\mathcal{O}(\max(n, N)^3)$, taking more snapshots will dramatically increase the computation time to obtain the reduced bases. For small numbers of snapshots ($\dim(\widehat{\mathcal{S}}) < 1000$) this may not be so bad, but if Branch-Ortho generates many thousands of snapshots (as may be the case for highly-branched cells) then the SVD step may become prohibitively slow.

Algorithm 3 : Branch-Ortho

Input: A set of snapshots $\mathcal{S}$
Output: Optimal set of routes $\widehat{\mathcal{R}}$ and set of "orthogonalized" snapshots $\widehat{\mathcal{S}}$

```
for $j = 1$ to $L$ do
  Let $n = 1$, $p_n \equiv$ the index of leaf $j$
  $p_{n+1} \equiv$ index of parent of branch $p_n$
  while ($p_{n+1} \notin \bigcup_{k=1}^{j-1} \mathsf{R}_k$) AND ($p_{n+1}$ is not the soma) do
    Increment $n$
    $p_{n+1} \equiv$ index of parent of branch $p_n$
  end while
  Route $\mathsf{R}_j = \{p_1, \ldots, p_n\}$
end for
Optimal set of routes is $\widehat{\mathcal{R}} = \bigcup_{j=1}^{L} \mathsf{R}_j$
for $j = 1$ to $L$ do
  $\mathcal{L}_j = \mathcal{S}$
  $\mathcal{B}_j = \{$indices of compartments that DO NOT belong to route $j\}$
  Set values of elements $\mathcal{B}_j$ of each snapshot in $\mathcal{L}_j$ to their rest values
end for
"Orthogonalized" set is $\widehat{\mathcal{S}} = \bigcup_{j=1}^{L} \mathcal{L}_j$
```

To keep the final snapshot set small while retaining the salient dynamics, we implement a snapshot elimination algorithm. A snapshot is "active" if its deviation from rest

$$\mathcal{D}(\mathrm{v}) \equiv \frac{1}{N} \sum_{j=1}^{N} |\mathrm{v}_j - \overline{\mathrm{v}}_j|^2 \tag{3.26}$$

exceeds a preset threshold. The deviation can be similarly defined for the snapshots of the nonlinear term. This process, called V-Slim, is detailed in Algorithm 4.

Algorithm 4 : V-Slim

```
Input: A set $\mathcal{S}$ of $m$ snapshots, and a tolerance $\varepsilon \in [0, 1)$
Output: The set $\widehat{\mathcal{S}}$ of active snapshots
for $j = 1 : m$, compute the deviation of the $j$th snapshot ($\mathcal{D}_j := \mathcal{D}(\mathcal{S}_j)$)
Let $M = \max_{j=1:m}(\mathcal{D}_j)$
Normalize deviations (set $\mathcal{D}_j := \mathcal{D}_j / M$ for all $j = 1 : m$)
$\mathcal{A} = \{j : \mathcal{D}_j < \varepsilon\}$
$\widehat{\mathcal{S}} = \{\mathcal{S}_j : j \in \mathcal{A}\}$
```

Another technique to shrink the size of $\widehat{\mathcal{S}}$ is to store only every kth snapshot in the local set $\mathcal{L}_j$. This is reasonable because snapshots that are close together in time are more similar. Keeping every snapshot gives maximal information, but we expect that we can get by with less (and our results indicate that this is indeed the case).

In our current implementation, such attention to shrinking the set of snapshots is necessary, but in the future one could use ARPACK (via the `eigs` command in MATLAB), which is a suite of iterative eigensolvers tailored to large-scale problems, to drastically reduce the cost of computing these SVDs. Previous experience gives us confidence that by using this package we can even eliminate the need for the aforementioned snapshot-shrinking algorithms. Nevertheless, the methods given here contribute a one-time cost, because results of these computations can be stored and reused for all future simulations of a specific morphology. This, along with the fact POD and DEIM set-up times for the simulations in this paper often required ~ 1 second and at most ~ 10 seconds, means we do not include set-up times in any of our timing comparisons.

3.3.2.4 Branch-Ortho and V-Slim Implementation

We can efficiently apply both Branch-Ortho and V-Slim if we consider the structure of the snapshots. The initial set $\mathcal{S}$ is likely to have a significant number of inactive snapshots, because of either delay due to onset of the stimulus or of the decay back to rest after hyperpolarization. We call the set of active snapshots from this initial set the *global active zone*, $\mathcal{S}_{\text{global}}$, since it defines when any part of the neuron is active. Applying Branch-Ortho to the global active zone will produce a smaller orthogonalized snapshot set than applying it to all of $\mathcal{S}$, so it is natural to use V-Slim on $\mathcal{S}$ to obtain $\mathcal{S}_{\text{global}}$ before applying Branch-Ortho.

Once the orthogonalized snapshot set is in hand, we apply V-Slim again to find the *local active zone* for each route, that is, the set of snapshots in $\mathcal{L}_j$ that are active for route j. The local active zone is often significantly smaller than $\mathcal{L}_j$ because an individual route is not active during all of the global active zone, but rather it could be inactive when other routes are active (due to travel time of the action potential or different dendritic lengths). Thus applying V-Slim to the orthogonalized snapshot set filters out these unnecessary local snapshots and can drastically reduce the size of the final snapshot set.

Thus our implementation of Branch-Ortho and V-Slim uses a 3-step process. First we apply V-Slim with a tolerance $\varepsilon_{\text{global}}$ to $\mathcal{S}$ to obtain the global active zone, $\mathcal{S}_{\text{global}}$. Next we run Branch-Ortho with $\mathcal{L}_j = \mathcal{S}_{\text{global}}$ to obtain $\widehat{\mathcal{S}}$. Finally, we apply V-Slim with a tolerance of $\varepsilon_{\text{local}}$ to $\widehat{\mathcal{S}}$, which effectively isolates the local active zones for each branch and yields the final snapshot set (see Figure 3.3). Note that since we have two snapshot sets, one for voltage and one for the nonlinear terms, the tolerances $\varepsilon_{\text{global}}$ and $\varepsilon_{\text{local}}$ can be different for each set.

3.3.3 Forked Neuron Results

Equipped with the tools from the above sections, we return to the task of reducing the forked neuron. We take 200 snapshots over a 10 ms window (using $\Delta t = 0.1$ ms), and we obtain them by giving a suprathreshold stimulus to a distal branch. We use the branchwise orthogonalization algorithm to generate snapshots of branches in isolation, where the active zone is computed by applying V-Slim with tolerances $\varepsilon^{v}_{\text{local}} = 0$, $\varepsilon^{f}_{\text{local}} = 0$, $\varepsilon^{v}_{\text{global}} = 10^{-6}$, and $\varepsilon^{f}_{\text{global}} = 10^{-5}$, and then saving every 4th snapshot from this resulting set. Simulations consist of 750 random step currents injected over a 1 second interval, with each current having amplitude 0–60 pA and lasting 0–5 ms.

Branch-Ortho turns out to be a very effective method of improving the accuracy of the reduced system, as Table 3.3 demonstrates. Not only are spike times accurately reproduced, but the somatic voltage traces are nearly exactly duplicated in the reduced system, as shown in Figure 3.5. The improvement can be seen qualitatively in Figure 3.4 by observing that the DEIM points are more evenly spaced throughout the neuron than they are without Branch-Ortho.

Our next test is done with the HHA model, but with all the other parameters

Table 3.3: Performance of reduced model (here $k_v = k_f$) of HH fork, $N = 1501$, as compared with the full model. The first simulation set does not use Branch-Ortho for the reduced system, while the next four sets use it, leading to far superior accuracy.

k_v	Speed-up	% Matched	% Mismatched	Γ
30, no Branch-Ortho	17.5×	34.9	19.0	0.484
10	24.1×	63.4	18.7	0.707
15	22.5×	91.5	4.5	0.933
20	21.2×	94.3	1.5	0.963
30	17.5×	99.6	0	0.998

Table 3.4: Performance of reduced model (here $k_v = k_f$) of HHA fork, $N = 1501$, as compared with the full model.

k_v	Speed-up	% Matched	% Mismatched	Γ
20	23.5×	48.27	41.42	0.515
25	22.0×	85.87	0	0.924
30	19.7×	98.9	2.6	0.981
40	15.4×	100	0	1

kept the same, except in this case, we use $\varepsilon^v_{\text{local}} = 10^{-4}$ and $\varepsilon^f_{\text{local}} = 10^{-5}$, and we use 1000 random stimuli of between 0–250 pA. Note that the tolerances here are not tuned to give the best performance, but rather have been chosen after just a little experimentation to give good performance; it is possible that a better choice may exist. Even though the spatially-varying A-type K^+ conductance necessitates a slightly larger reduced system in order maintain the accuracy seen in the HH fork, we still observe a similar speed-up, as Table 3.4 indicates.

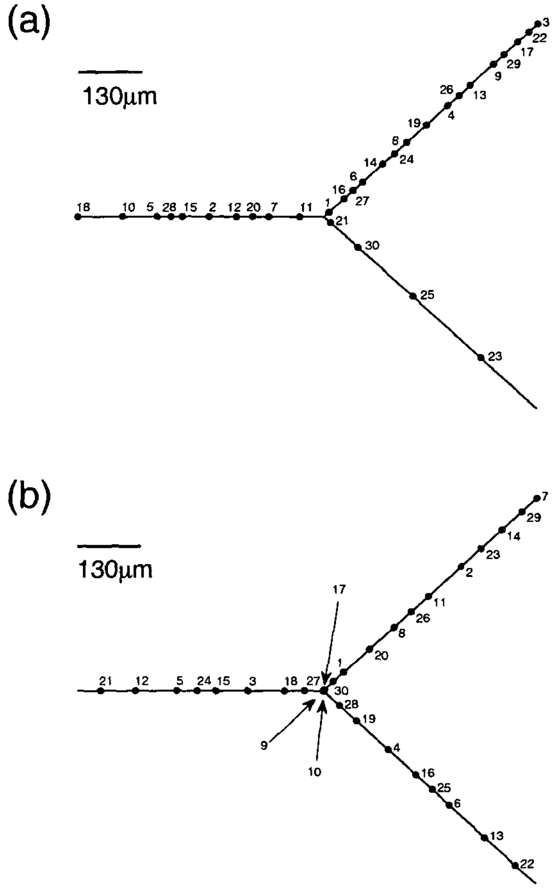


Figure 3.4: DEIM points computed for the HH forked neuron when branchwise orthogonalization is (a) not used and (b) used. The improvement from using Branch-Ortho is evident in the qualitative difference in DEIM point location: notice that in (a) the lower-right branch has few DEIM points, whereas in (b) the DEIM points are more evenly distributed. In (b) there is a small cluster of points near the junction, with each point lying on distinct branches.

3.3.4 Branched Cell Obeying Rall's 3/2 Power Law

To examine more complex branching patterns we now consider a binary branching tree which satisfies the 3/2 power law as explained in (Rall, 1959). The cell has a depth of 3 (for a total of 15 branches), and the root branch has length 200 μm and

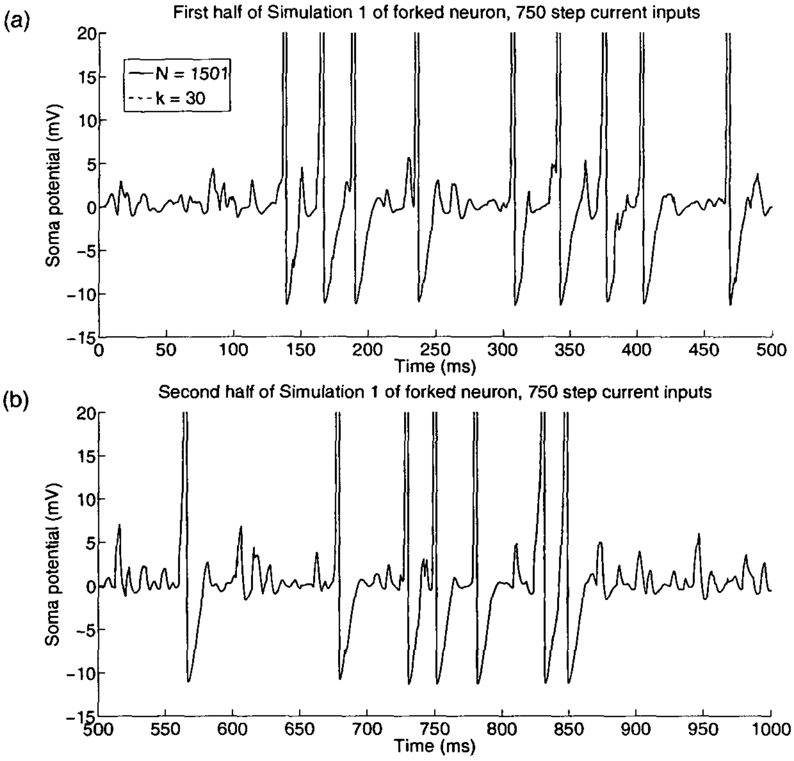


Figure 3.5: Somatic voltage traces of the nonlinear and reduced models for (a) first half and (b) second half of simulation 1 of the HH forked neuron.

radius 8 μm, agreeing with the morphological parameters as given in Rallpack 2 of (Bhalla et al., 1992). We consider two variants of such a tree: one in which the dendritic radii do not taper and one in which they taper linearly along the length of the dendrites. In the reduced systems, the V-Slim tolerances are $\varepsilon^v_{\text{local}} = 10^{-2}$, $\varepsilon^f_{\text{local}} = 10^{-3}$, $\varepsilon^v_{\text{global}} = 10^{-6}$, and $\varepsilon^f_{\text{global}} = 10^{-5}$, and the step between snapshots in the global active zone is 4. We run 20 simulations of 1000 ms each with $\Delta t = 0.1$ ms, just as for the forked neuron, only this time we use 600 random currents of 0–500 pA amplitude and 0–5 ms duration.

The results shown in Table 3.5 show that the reduced system recovers spiking

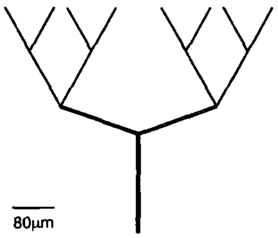


Figure 3.6: Rendering of a binary branching cell of depth 3 obeying Rall's 3/2 power law. The root branch has length 200 μm and radius 8 μm, agreeing with the morphological parameters as given in Rallpack 2 of (Bhalla et al., 1992).

Table 3.5: Performance of reduced model (here $k_v = k_f$) of HH Rall cells of depth 3 for tapered and non-tapered branches as compared with the full model ($N = 1823$).

k_v	Speed-up	% Matched	% Mismatched	Γ
30, tapered	19.0×	95.9	1.1	0.974
30, non-tapered	18.9×	79.2	8.9	0.836
40, non-tapered	14.8×	91.5	5.6	0.847
50, non-tapered	12.0×	93.9	5.4	0.874

dynamics to high accuracy. The tapered cell is less excitable than the non-tapered cell (which may be expected since there are more compartments with greater surface areas in the tapered cell). Also, the non-tapered cell requires a larger reduced system to capture the dynamics, which suggests that cells with smoothly varying dendritic radii may yield better results than cells whose radii change abruptly.

3.4 Results on Realistic Morphologies

3.4.1 Results on Neuron AR-1-20-04-A

Consider the projection interneuron AR-1-20-04-A (Martinez), which has 35 branches and 20 leaves. At 1 μm resolution, the full system has $N = 2233$ compartments.

We conduct a set of 20 simulations of 1 second duration, each using 500 random current injections of amplitudes between 0–150 pA and lasting 0–5 ms. We compare the spike-capturing accuracy of the full system to that of reduced systems of different dimensions. In the reduced systems, the V-Slim tolerances are $\varepsilon^{v}_{\text{local}} = 0.002$, $\varepsilon^{f}_{\text{local}} = 0.0005$, $\varepsilon^{v}_{\text{global}} = 10^{-6}$, and $\varepsilon^{f}_{\text{global}} = 10^{-5}$, and the step between snapshots in the global active zone is 4. The application of Branch-Ortho leads to highly accurate and fast reduced systems, as Figure 3.7 and Table 3.6 show.

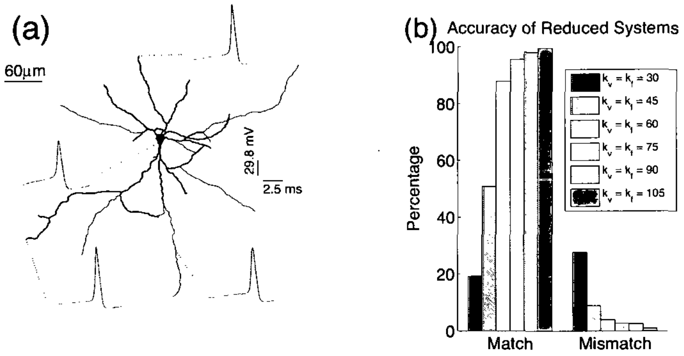


Figure 3.7: (a) Rendering of cell AR-1-20-04-A (Martinez) with proximal and distal voltage traces computed from the full and reduced systems (blue and red lines, respectively) in response to a 500 pA step current stimulus lasting 9 ms applied at the soma. The reduced system size for this plot is $k_v = k_f = 90$. (b) Bar plot showing the percentage matched and mismatched for the reduced systems for cell AR-1-20-04-A.

3.4.2 Results on Other Morphologies

The POD and DEIM modeling framework is applicable to a wide class of neurons. We have assembled a broad test group of neurons from (http://NeuroMorpho.org) (Ascoli, 2006). In Table 3.7 we summarize the accuracy and speed of the correspond-

Table 3.6: Results of model reduction for neuron AR-1-20-04-A, $N = 2233$ (here $k_v = k_f$), 500 step currents, HHA model.

k_v	Speed-up	% Matched	% Mismatched	Γ
30	29.0×	19.2	27.7	0.299
45	19.7×	50.8	8.9	0.650
60	14.8×	87.9	4.0	0.916
75	9.6×	95.5	2.7	0.963
90	6.9×	98.1	2.6	0.977
105	5.3×	99.5	0.8	0.992

ing reduced systems, and we plot the coincidence factors for a range of reduced system sizes in Figure 3.8. In general, the reduced systems are successful at capturing the spiking dynamics of each cell, and they simulate faster than their full system counterparts. Additionally, the voltage dynamics at any location can be recovered from the reduced system, not just the somatic potential (for example, see Figure 3.7).

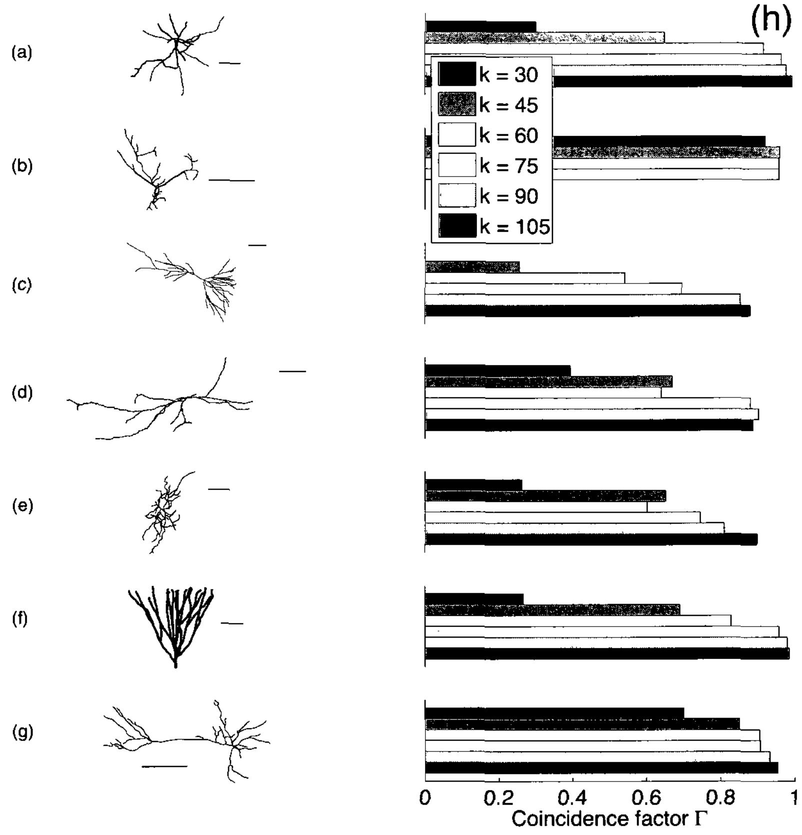


Figure 3.8: Coincidence factors Γ show spike capturing accuracy for seven different cells using various sizes of reduced systems ($k = k_v = k_f$) and Algorithm 3 (see (h)). The strongly excitable channel models are used here (see Table 3.7 for the model used on each cell). From top to bottom the cells are: (a) projection interneuron AR-1-20-04-A (Martinez), (b) CA3 hippocampal interneuron 951005a (Chitwood et al., 1999), (c) CA3 hippocampal pyramidal cell 12299402 (Chitwood et al., 1999), (d) CA1 hippocampal interneuron 100103a from the stratum oriens (Golding et al., 2001) (Golding et al., 2005), (e) retinal ganglion cell mp_tb_40984_gc1 (Toris et al., 1995), (f) hippocampal cell 512882 from the dentate gyrus (Rihn and Claiborne, 1990), and (g) pyramidal cell P8-DEV66 from the cerebral cortex (Furtak et al., 2007). Scale bars on all cells represent 100 μm.

Table 3.7: Specifications and performance of reduced systems ($k_v = k_f$) for the morphologies shown in Figure 3.8 using the HH or HHA channel models. Coincidence factors shown correspond to using Algorithm 3. Here $\mathcal{B}$ = number of branches in each cell, and for each cell we set $h = 1$ μm to be the desired compartment size.

Cell	Model	$\mathcal{B}$	N	k_v	N/k_v	Speed-up	Γ
AR-1-20-04-A	HHA	35	2233	75	29.8×	9.6×	0.963
951005a	HHA	44	1106	45	24.6×	10.0×	0.960
12299402	HH	61	5021	105	47.8×	9.2×	0.880
100103a	HHA	32	2707	90	30.1×	7.3×	0.903
mp_tb_40984_gc1	HH	54	2541	105	24.2×	4.7×	0.899
512882	HH	35	4655	90	51.7×	11.5×	0.981
P8-DEV66	HH	47	1712	60	28.5×	9.8×	0.905

3.4.3 Cells With Weakly Excitable Dendrites

Up to this point we have considered ion channel distributions (HH and HHA) that lead to strongly excitable dendrites. However, real neurons often have weakly excitable dendrites due in large part to the increase of the density of K^+ channels with distance from the soma. In order to assess how well the POD and DEIM capture the spiking dynamics of these weakly excitable cells, we use the MIG channel model (see Table B.3) with the previously considered morphologies.

In addition to considering the full model versus the reduced model, we also offer a comparison of the accuracy of coarsened models. That is, we use larger values of h for the full system and compare the resulting spike trains to those computed by the full system using the fine reference value of $h = 1$ μm. As shown in Table 3.8, coarsening of cell AR-1-20-04-A yields less accurate results than the reduced systems, and the reduced systems are also much faster. This presents a strong argument in

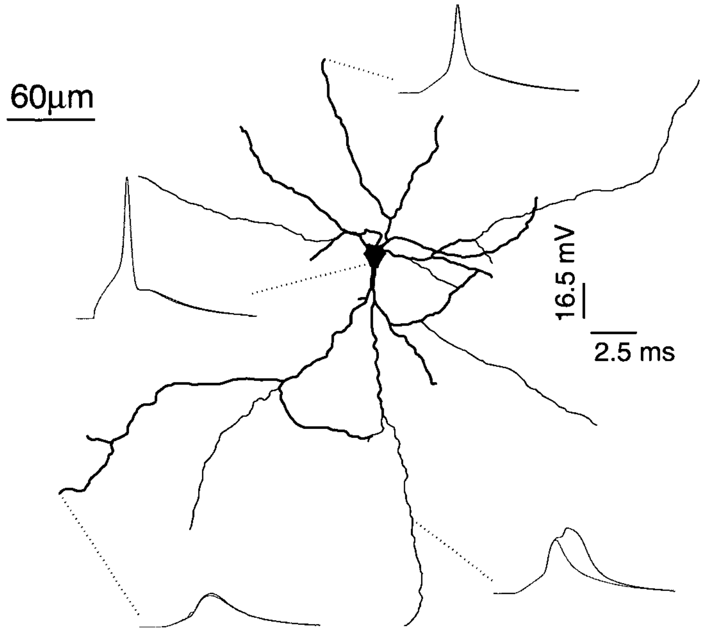


Figure 3.9: Voltage traces at various spatial locations along cell AR-1-20-04-A for the full and reduced systems (blue and red lines, respectively) in response to a 1000 pA step current stimulus lasting 2 ms applied to the soma. The traces show that accurate reduced systems can be generated even when using ion channel models that yield weakly excitable dendrites. Here we use the MIG channel model and a reduced system of size $k_v = k_f = 90$ and the time-step is $\Delta t = 0.01$ ms. The locations in this figure are the same as those of Figure 3.7a.

favor of using the model reduction techniques presented here over the current standard method of coarsening for simulations of single neurons.

Just as in the previous section, in Table 3.9 we summarize the accuracy and speed of the reduced systems for these cells with the MIG channel model, and we plot the coincidence factors for a range of reduced system sizes in Figure 3.10. Though the accuracy is generally lower than for cells with the HH and HHA models, the reduced

Table 3.8: Accuracy of coarsened full systems (top section) and reduced systems (bottom section) for the projection interneuron AR-1-20-04-A with the MIG channel model. Simulations were conducted using 2000 random current injections of duration and magnitude as in §3.4.1, and $k_v = k_f$ for the reduced systems.

h (μm)	Speed-up	% Matched	% Mismatched	Γ
5	4.6×	74.1	49.2	0.595
2	2.0×	77.9	42.4	0.656
k_v				
60	12.3×	87.8	22.3	0.821
75	8.1×	88.6	21.6	0.829
90	5.9×	82.4	12.2	0.849
105	4.3×	93.9	10.2	0.917

Table 3.9: Specifications and performance of reduced systems ($k_v = k_f$) for the morphologies shown in Figure 3.10 using the MIG channel model. Cell mp_tb_40984_gc1 is marked with a '*' because the reduced system became unstable before k_v was large enough to capture any significant dynamics, thus this was the only cell for which the reduced system failed for the MIG channel model.

Cell	Model	$\mathcal{B}$	N	k_v	N/k_v	Speed-up	Γ
AR-1-20-04-A	MIG	35	2233	120	18.6×	3.3×	0.933
951005a	MIG	44	1106	75	14.7×	4.3×	0.893
12299402	MIG	61	5021	150	33.4×	4.6×	0.882
100103a	MIG	32	2707	120	22.5×	4.0×	0.886
*mp_tb_40984_gc1	MIG	54	2541	60	42.4×	14.7×	0.471
512882	MIG	35	4655	105	44.3×	9.2×	0.928
P8-DEV66	MIG	47	1712	75	22.8×	6.7×	0.856

systems are still successful at capturing the spiking dynamics of most cells. However, larger reduced systems must be used to resolve these dynamics, so smaller speed-ups are observed. The MIG channel model also shows us a case of utter failure for the first time: for the retinal ganglion cell in Figure 3.10e, we find that numerical instabilities, most likely due to poor snapshots, in the reduced system occur for larger dimensions.

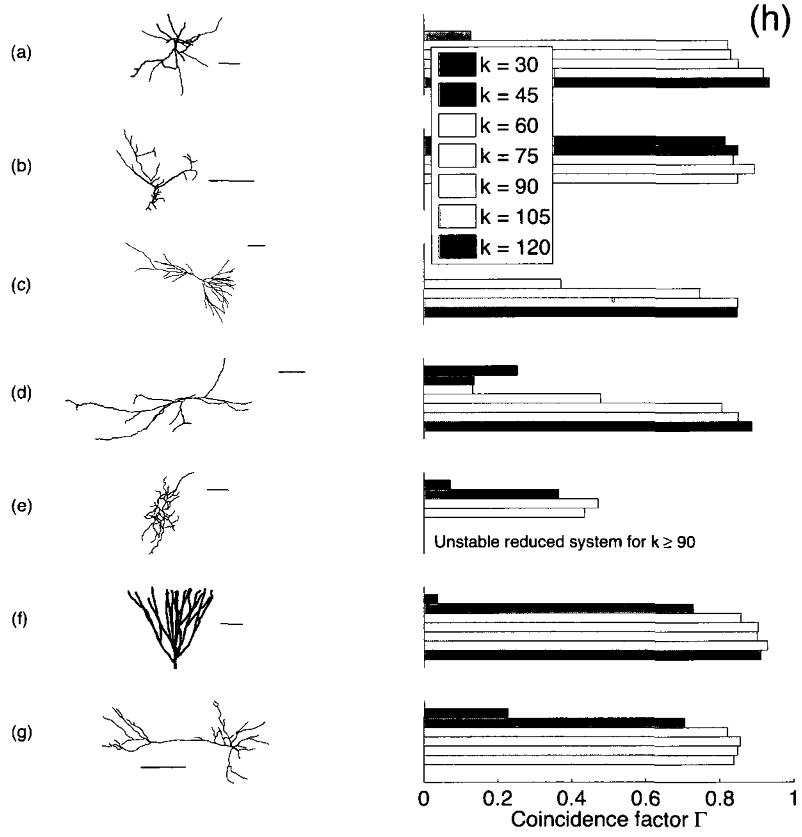


Figure 3.10: Coincidence factors $\boldsymbol{\Gamma}$ show spike capturing accuracy for seven different cells using various sizes of reduced systems ($k = k_v = k_f$) and Algorithm 3 (see (h)) using the weakly excitable channel model (see Table 3.9). Cells are the same as those of Figure 3.8, and scale bars on all cells represent 100 μm.

3.5 Synaptic Conductances vs. Current Injection

The reduced system can handle synaptic conductances and current injections, and they can be used interchangeably (i.e., a set of snapshots generated using current injection can also be used for a simulation with synaptic conductances). But, synaptic conductances take a computational toll on the reduced model because, in order to maintain an implicit time-stepping scheme, (3.21) implies that we must compute $\mathtt{U}^T\text{diag}(\mathtt{G}(t))\mathtt{U}$, where $\mathtt{G}(t)$ is the input synaptic conductance to each compartment. When many synaptic inputs are present, this becomes an expensive double-matrix product which can dominate the total simulation time. If alpha functions are used to model synaptic events, we can decrease the number of active inputs via a shutoff mechanism, thus greatly accelerating this computation. In fact, shutoff mechanisms can be analogously implemented for any decaying synaptic conductance time course.

Alpha functions have a characteristic form

$$\tilde{g}(t) = \overline{g}\frac{t - t_{\text{on}}}{\tau}\exp\left(1 - \frac{t - t_{\text{on}}}{\tau}\right) \tag{3.27}$$

where τ is the time constant and $\overline{g}$ is the maximal conductance. Now we assume that the synaptic event is inactive after some shutoff time, t_{off}, at which $\tilde{g}(t_{\text{off}}) = \varepsilon$ and $\tilde{g}'(t_{\text{off}}) < 0$ for some small tolerance ε. The truncated alpha function can then be implemented as

$$g(t) = \begin{cases} \tilde{g}(t), & t_{\text{on}} \leq t < t_{\text{off}} \\ 0, & \text{otherwise.} \end{cases} \tag{3.28}$$

Shutoff mechanisms can significantly speed up simulations when many synaptic inputs are used. For example, for cell AR-1-20-04-A with $N = 2233$, $k_v = k_f = 60$, and 500 inputs with $\overline{g} = 2$ nS and $\tau = 1$, we observe a speed-up in calculating the double-matrix product from 30 seconds to 3.3 seconds when the shutoff tolerance $\varepsilon = 10^{-4}$ is used. Now the reduced system is much more competitive with the full system, as Table 3.10 illustrates. Compared to Table 3.6 the accuracy and speed-ups in Table 3.11 are less, due to the fact that synaptic input is more complex than current injection, but the reduced system is still very successful at capturing the cell's behavior.

Table 3.10: Alpha synapse shutoff mechanism ($\varepsilon = 10^{-4}$) accelerates the reduced system of neuron AR-1-20-04-A.

		Sim. Time (sec)	
Type	Dimension	Without shutoff	With shutoff
Full	$N = 2233$	115	115
Reduced	$k_v = k_f = 60$	41	15
Reduced	$k_v = k_f = 90$	73	32

Table 3.11: AR-1-20-04-A, $N = 2233$ (here $k_v = k_f$), 500 alpha synapses, HHA model.

k_v	Speed-up	% Matched	% Mismatched	Γ
60	8.4×	86.6	22.8	0.807
75	5.9×	92.8	12.8	0.894
90	4.7×	93.9	15.1	0.886
105	3.6×	98.6	6.8	0.955
120	2.9×	98.6	5.9	0.961

3.6 Discussion

We have applied nonlinear model reduction techniques to morphologically realistic cells in a way that preserves the input-output relationships while accurately reproducing the complete voltage dynamics. We approximate the voltage using a POD basis, which reduces the number of state variables. Using the DEIM we build a set of spatial interpolation points and basis vectors to reduce the complexity of the nonlinear term so that it is proportional to the number of reduced variables. These techniques rely upon having "snapshots" of the dynamics that are rich enough to reproduce the full range of neuronal behavior. We introduce simple methods for obtaining these snapshots, and we develop algorithms to eliminate unimportant snapshots and to maximize the information in the important ones. When these tools are applied to realistic cells, they can generate reduced systems of dimensions nearly two orders of magnitude smaller than the originals and which yield highly accurate simulations in an order of magnitude less time.

Application of these model reduction techniques is likely to have the greatest effect on neuronal models with realistic ionic currents. Our simulations currently use standard, but limited, channel kinetics. Behaviors such as action potential initiation near the soma (Colbert and Pan, 2002) (Kole et al., 2008) rely upon better models for the spatial conductance densities, as well as the inclusion of different channels (Mainen and Sejnowski, 1998). Additionally, in this paper we have only considered excitatory inputs, but inhibition plays a key role in modulating neuronal output, both at the single cell and network levels (Brunel and Wang, 2003). Inhibition can be included with no change to the present methodology.

An improvement of the model reduction technique can be made when it is applied to cells with weakly excitable dendrites. One consequence of using such channel models is that some distal branches do not experience action potentials, but rather have more attenuated responses, which can result in poor snapshot dynamics. A remedy for this is to augment the snapshot set with extra snapshots generated by giving inputs to these "dead branches" in order to more accurately capture their dynamics. We have found this to yield significant improvements for some cells, though our selection of the branches to stimulate has been ad hoc; developing an automated algorithm for this process will be part of our future work.

Synaptic conductances can also be treated via AMPA, GABA, and NMDA receptors, and their implementation in the reduced model is straightforward. The DEIM can even be used to reduce the dimension of the voltage-dependent NMDA receptor, as we have done in preliminary codes.

Chapter 4

The Reduced Strong-Weak Model

In the previous two chapters I introduced two different techniques to reduce the dimension of neuronal models, and in this chapter I will fuse them together in a model that uses both techniques where they are most advantageous. If the ion channel models were purely linear (that is, if the dendrites were purely passive), then the linear reduction techniques would be able to reproduce to multi-digit accuracy the somatic voltage. As explained in §3.4.3, many real cells have weakly excitable distal dendrites, and these branches behave like passive cables, whereas the proximal dendrites are active cables. It makes sense, therefore, to consider a model in which the weakly excitable dendrites are modeled as a linear system, while the strongly excitable dendrites are modeled as a nonlinear system, and then to apply the corresponding reduction techniques to each part.

This reduced strong-weak model, while not a cure-all for neuronal simulations, opens doors to accurate reduced simulation of certain cells that otherwise would be out of reach. Since the linear reduction techniques work when there are only a few observables, we do not aim to accurately reduce cells with non-tufted or weblike branching patterns, such as Purkinje neurons (Stuart et al., 2008). However, cells such as pyramidal neurons with branching patterns that are organized into two distinct dendritic tufts are better suited for success with the reduced strong-weak model.

Here I derive the strong-weak model and its reduction, and I demonstrate that it improves upon the spike-capturing accuracy of the reduced model that uses only the POD and DEIM techniques. First I begin with the derivation for a uniform fiber in order to more clearly show the changes that occur, and then I generalize the derivation to apply to any morphology.

4.1 Constructing the Strong-Weak Model

I begin with the simplest morphology, a uniform fiber of radius a and length ℓ. Using an ion channel model that yields weakly excitable distal dendrites (see Table B.3), we identify the *transition location* x_T on the fiber at which we say the voltage behavior transitions from strong (active) to weak (quasi-active). The method to identify x_T is to be determined, but once we have it we can partition the fiber into two separate fibers, one strong (from $x = 0$ to $x = x_T$) and one weak (from $x = x_T$ to $x = \ell$), which join at the point x_T, as shown in Figure 4.1.

The absolute voltages of each fiber will be denoted V_s and V_w, while the voltages with respect to rest will be denoted v_s and v_w. Similarly, V_T and v_T will denote the

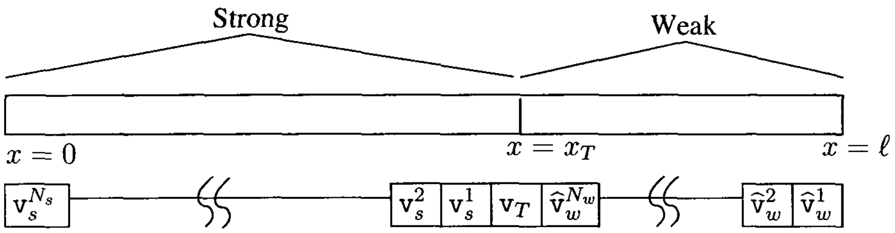


Figure 4.1: Schematic of the strong-weak model for a uniform fiber, along with its spatial discretization to illustrate the ordering of compartments

absolute and relative voltages at the transition point:

$$V_T(t) = V_s(x_T, t) = V_w(x_T, t), \quad v_T(t) = V_T(t) - \overline{V}_T. \tag{4.1}$$

It will be computationally advantageous to work with voltages with respect to rest.

The strong fiber obeys the nonlinear cable equation, while the weak fiber obeys the quasi-active version of this PDE. Hence, using the cable equation from §2.3.1, we write the strong-weak system as

$$aC_m\partial_t v_s = \frac{1}{2R_i}a^2\partial_{xx}v_s - a\sum_{c=1}^{C} G_{sc}(x)(v_s - E_c)\prod_{f=1}^{F_c} w_{scf}^{q_{cf}} - I_s(v_s, x, t) \tag{4.2}$$

$$\partial_t w_{scf} = \frac{w_{cf,\infty}(v_s) - w_{scf}}{\tau_{cf}(v_s)}, \qquad 0 < x < x_T,\ 0 < t, \tag{4.3}$$

$$\partial_t \begin{bmatrix} \widehat{v}_w \\ \widehat{w}_{w,:} \end{bmatrix} = \mathcal{Q}_w\left(\begin{bmatrix} \widehat{v}_w \\ \widehat{w}_{w,:} \end{bmatrix}\right) + \begin{bmatrix} \widehat{I}_w(\widehat{v}_w, x, t) \\ 0 \end{bmatrix}, \qquad x_T < x < \ell,\ 0 < t, \tag{4.4}$$

where $\widehat{v}_w$ and $\widehat{w}_{w,:}$ denote the linearized weak voltage and gating variables, Φ denotes the ionic current function, I_s and denotes the inputs to the strong fiber, $\widehat{I}_w$ denotes the linearized inputs to the weak fiber, and $\mathcal{Q}_w$ is the quasi-active operator, as in equation (2.32), which is applied to the weak voltages and gating variables.

The initial condition for this system is that the two fibers are at rest, and this rest potential is identical to that of the full fiber. The sealed end conditions imply that

$$\partial_x v_s(0,t) = \partial_x \widehat{v}_w(\ell,t) = 0. \tag{4.5}$$

Continuity of voltage at the transition point implies

$$v_s(x_T,t) = \widehat{v}_w(x_T,t) \tag{4.6}$$

and continuity of current at the transition point implies that

$$\partial_x v_s(x_T,t) = \partial_x \widehat{v}_w(x_T,t). \tag{4.7}$$

Now we discretize this model. Assume that we are given a compartment length Δx, and further assume that the transition compartment (the one containing x_T) belongs to neither the strong fiber nor the weak one, but is rather shared by both. Then there are N_s and N_w compartments in the strong and weak fibers, respectively, and the total number of compartments in the full fiber is $N = N_s + N_w + 1$. Indexing the compartments from the outside in for each fiber implies that discretizing (4.7) yields

$$\frac{\mathrm{v}_s^1 - \mathrm{v}_T}{\Delta x} = \frac{\mathrm{v}_T - \widehat{\mathrm{v}}_w^{N_w}}{\Delta x}$$

where v_T is the voltage at the transition compartment. By (4.6) this implies

$$\mathrm{v}_T = \frac{\mathrm{v}_s^1 + \widehat{\mathrm{v}}_w^{N_w}}{2}. \tag{4.8}$$

Although the implementation above is only first-order accurate, it can be improved to second-order accuracy as described in §A.2. However, all the results in this thesis were produced using the first-order scheme.

Discretizing the ∂_{xx} terms in the above systems and substituting in for v_T we arrive at expressions for the voltages at the compartments adjacent to the transition compartment:

$$\begin{aligned}
\partial_{xx}\mathrm{v}_s^1 &\approx \frac{\mathrm{v}_s^2 - 2\mathrm{v}_s^1 + \mathrm{v}_T}{(\Delta x)^2} \\
&= \frac{\mathrm{v}_s^2 - 2\mathrm{v}_s^1 + \frac{1}{2}\mathrm{v}_s^1 + \frac{1}{2}\widehat{\mathrm{v}}_w^{N_w}}{(\Delta x)^2} \\
&= \frac{\mathrm{v}_s^2 - \frac{3}{2}\mathrm{v}_s^1 + \frac{1}{2}\widehat{\mathrm{v}}_w^{N_w}}{(\Delta x)^2} \\
\partial_{xx}\widehat{\mathrm{v}}_w^{N_w} &\approx \frac{\frac{1}{2}\mathrm{v}_s^1 - \frac{3}{2}\widehat{\mathrm{v}}_w^{N_w} + \widehat{\mathrm{v}}_w^{N_w-1}}{(\Delta x)^2}.
\end{aligned}$$

This reveals the coupling between these two systems. Using the Hines ordering, I now discretize (4.2)–(4.4) and the conditions given in (4.5) and (4.7) to arrive at

$$\partial_t \begin{bmatrix} \widehat{\mathtt{v}}_w \\ \widehat{\mathtt{w}}_w \\ C_m \mathtt{v}_s \end{bmatrix} = c \left[\begin{array}{c|ccccc} \mathtt{Q}/c & (1/2)e_{N_w} & 0 & \cdots & & 0 \\ \hline (1/2)e_{N_w}^T & -3/2 & 1 & & & \\ 0 & 1 & -2 & 1 & & \\ \vdots & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 0 & & & & 1 & -1 \end{array}\right] \begin{bmatrix} \widehat{\mathtt{v}}_w \\ \widehat{\mathtt{w}}_w \\ C_m \mathtt{v}_s \end{bmatrix}$$

$$- \begin{bmatrix} 0 \\ 0 \\ \big(\Phi(\mathtt{w}_s(t))\mathtt{e}\big).(\mathtt{v}_s(t) + \overline{\mathtt{v}}_s) - \Phi(\mathtt{w}_s(t))\mathtt{E}_i \end{bmatrix} \tag{4.9}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ \big(\Phi(\overline{\mathtt{w}}_s)\mathtt{e}\big).\overline{\mathtt{v}}_s(t) - \Phi(\overline{\mathtt{w}}_s)\mathtt{E}_i \end{bmatrix}$$

$$+ \frac{1}{2\pi a \Delta x} \begin{bmatrix} \mathtt{B}\widehat{\mathtt{I}}_w(t) \\ \mathtt{I}_s(t) \end{bmatrix},$$

$$\mathtt{w}_s'(t) = \big(\mathtt{A}(\mathtt{v}_s(t) + \overline{\mathtt{v}}_s) - \mathtt{w}_s(t)\big)./\mathtt{B}(\mathtt{v}_s(t) + \overline{\mathtt{v}}_s), \tag{4.10}$$

where $\mathtt{v}_s \in \mathbb{R}^{N_s}$, $\widehat{\mathtt{v}}_w \in \mathbb{R}^{N_w}$, and $\widehat{\mathtt{w}}_w \in \mathbb{R}^{N_w m}$, and where $\mathtt{Q} \in \mathbb{R}^{N_w(m+1)\times N_w(m+1)}$ is the quasi-active system matrix, $\mathtt{B} = \begin{bmatrix} I/C_m & 0 \end{bmatrix}^T \in \mathbb{R}^{N_w(m+1)\times N_w}$ is the quasi-active input matrix, $c = \frac{a}{2R_a(\Delta x)^2}$, and where $e_{N_w} \in \mathbb{R}^{N_w(m+1)}$ is the coordinate vector corresponding to N_w. The definitions of the ionic term Φ and the strong gating variables $\mathtt{w}_s$ follow as in §3, just adjusted to the proper number of strong compartments.

Notice that the nonlinear term Φ has been split into two parts, one at which to evaluate the ionic currents at the absolute strong voltage and one at the resting strong voltage. This is because the state variables are with respect to rest, but Φ takes absolute voltage values, implying that the ionic term is not computed with respect to rest, unlike the other quantities in the system. This nonlinear ionic contribution must also be adjusted to be with respect to rest, and including the resting ionic term

$$\overline{\Phi} = \big(\Phi(\overline{\mathtt{w}}_s)\mathtt{e}\big).\overline{\mathtt{v}}_s(t) - \Phi(\overline{\mathtt{w}}_s)\mathtt{E}_i \tag{4.11}$$

achieves this goal.

Also note that the transition compartment is not explicitly represented in (4.9). Rather, in order to obtain $\mathtt{v}_T$, equation (4.8) must be used. One implication of this compartment being left out is that any inputs which would arrive here are ignored. To remedy this, the present solution is to shift the input into one of the adjacent compartments.

Defining the matrices

$$\begin{aligned}
\mathtt{Z}_w &= c\frac{1}{2}e^T_{N_w} \in \mathbb{R}^{1\times N_w(m+1)} \\
\mathtt{Z}_s &= \mathtt{Z}_w^T/C_m \\
\mathtt{H}_s &= c\begin{bmatrix} -3/2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{N_s\times N_s},
\end{aligned}$$

we write the coupled strong-weak "Hines" matrix as

$$\mathtt{H} = \left[\begin{array}{c|cc} \mathtt{Q} & \mathtt{Z}_s & 0 \\ \hline \mathtt{Z}_w & \mathtt{H}_s & \\ 0 & & \end{array}\right] \tag{4.12}$$

and for convenience we define

$$\mathtt{N}(\mathtt{v}_s(t),\mathtt{w}_s(t)) \equiv \big(\Phi(\mathtt{w}_s(t))\mathtt{e}\big).(\mathtt{v}_s(t)+\overline{\mathtt{v}}_s) - \Phi(\mathtt{w}_s(t))\mathtt{E}_i \tag{4.13}$$

which, along with (4.11), simplifies (4.9) to

$$\partial_t \begin{bmatrix} \widehat{\mathtt{v}}_w \\ \widehat{\mathtt{w}}_w \\ C_m\mathtt{v}_s \end{bmatrix} = \mathtt{H} \begin{bmatrix} \widehat{\mathtt{v}}_w \\ \widehat{\mathtt{w}}_w \\ C_m\mathtt{v}_s \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \mathtt{N}(\mathtt{v}_s(t),\mathtt{w}_s(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \overline{\Phi} \end{bmatrix} + \frac{1}{2\pi a\Delta x}\begin{bmatrix} \mathtt{B}\widehat{\mathtt{I}}_w(t) \\ \mathtt{I}_s(t) \end{bmatrix}. \tag{4.14}$$

4.1.1 Model Reduction of the Weak Part

We focus first on reducing the weak part of the model by applying the linear model reduction techniques of §2. In order to proceed with the model reduction we need to write the weak part in the form of a standard linear system, but the coupling term from $\mathtt{Z}_s$ causes a problem. We make the assumption that this term can really be treated as an input, i.e., we assume that $\mathtt{v}_s^1$ is an input to the linear part, rather than treating it as a state variable.

With this assumption on the inputs, let $\mathtt{z}_w = \begin{bmatrix} \widehat{\mathtt{v}}_w ; & \widehat{\mathtt{w}}_w \end{bmatrix}^T$ be the state vector for the weak part. Then the linear system whose observable is the voltage $\mathtt{v}_w^{N_w}$ at the weak compartment adjacent to the transition compartment is

$$\begin{aligned} \mathtt{z}'(t) &= \mathtt{Q}\mathtt{z}(t) + \begin{bmatrix} \mathtt{B} & \mathtt{Z}_s \end{bmatrix} \begin{bmatrix} \widehat{\mathtt{I}}_w(t) \\ \mathtt{v}_s^1(t) \end{bmatrix} \\ \mathtt{v}_w^{N_w}(t) &= \mathtt{C}\mathtt{z}(t) \end{aligned} \tag{4.15}$$

where $\mathtt{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{1 \times N_w(m+1)}$. To compute a reduced system of dimension k_w, we apply either BT or IRKA to (4.15) to obtain

$$\begin{aligned} \boldsymbol{\xi}'(t) &= \tilde{\mathtt{Q}}\boldsymbol{\xi}(t) + \begin{bmatrix} \tilde{\mathtt{B}} & \tilde{\mathtt{Z}}_s \end{bmatrix} \begin{bmatrix} \widehat{\mathtt{I}}_w(t) \\ \mathtt{v}_s^1(t) \end{bmatrix} \\ \mathtt{v}_w^{N_w}(t) &\approx \tilde{\mathtt{C}}\boldsymbol{\xi}(t) \end{aligned}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{k_w}$, $\tilde{\mathtt{Q}} \in \mathbb{R}^{k_w \times k_w}$, $\tilde{\mathtt{B}} \in \mathbb{R}^{k_w \times N_w}$, $\tilde{\mathtt{Z}}_s \in \mathbb{R}^{k_w \times 1}$, and $\tilde{\mathtt{C}} \in \mathbb{R}^{1 \times k_w}$.

We insert this reduced system back into the coupled system of (4.14). Now we assume that the matrix $\tilde{\mathtt{Z}}_s$ can be applied to the state variable $\mathtt{v}_s^1$ directly, and hence we obtain the coupled, weakly-reduced system

$$\partial_t \begin{bmatrix} \boldsymbol{\xi} \\ C_m \mathtt{v}_s \end{bmatrix} = \left[\begin{array}{c|c} \tilde{\mathtt{Q}} & \tilde{\mathtt{Z}}_s \quad 0 \\ \hline (c/2)\tilde{\mathtt{C}} & \mathtt{H}_s \\ 0 & \end{array}\right] \begin{bmatrix} \boldsymbol{\xi} \\ C_m \mathtt{v}_s \end{bmatrix} - \begin{bmatrix} 0 \\ \mathtt{N}(\mathtt{v}_s(t), \mathtt{w}_s(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ \Phi \end{bmatrix} + \frac{1}{2\pi a \Delta x} \begin{bmatrix} \tilde{\mathtt{B}}\widehat{\mathtt{I}}_w(t) \\ \mathtt{I}_s(t) \end{bmatrix}. \tag{4.16}$$

4.1.2 Model Reduction of the Strong Part

With the weak part reduced, we now proceed to apply the POD and DEIM to reduce the strong part following the procedure as detailed in §3. First we take n snapshots of $\mathtt{v}_s$ and store them as

$$\mathtt{V} = \begin{bmatrix} \mathtt{v}_s(t_1) & \cdots & \mathtt{v}_s(t_n) \end{bmatrix} \in \mathbb{R}^{N_s \times n}.$$

We take the SVD of this matrix, $\mathtt{V} = U\Sigma X^T$, and choose a strong reduced system size $k_s \leq N_s$. The POD matrix is then the first k_s columns of U, i.e. $\mathtt{U} = U(:, 1 : k_s)$. Now we define the reduced variable by

$$\mathtt{v}_s = \mathtt{U}\tilde{\mathtt{v}}_s, \quad \tilde{\mathtt{v}}_s \in \mathbb{R}^{k_s}, \quad \mathtt{U} \in \mathbb{R}^{N_s \times k_s},$$

and then we substitute into (4.16) to obtain

$$\partial_t \begin{bmatrix} \boldsymbol{\xi} \\ C_m\mathtt{U}\mathtt{v}_s \end{bmatrix} = \left[\begin{array}{c|cc} \tilde{\mathtt{Q}} & \tilde{\mathtt{Z}}_s & 0 \\ \hline (c/2)\tilde{\mathtt{C}} & \mathtt{H}_s & \\ 0 & & \end{array}\right] \begin{bmatrix} \boldsymbol{\xi} \\ C_m\mathtt{U}\mathtt{v}_s \end{bmatrix} - \begin{bmatrix} 0 \\ \mathtt{N}(\mathtt{U}\mathtt{v}_s(t), \mathtt{w}_s(t)) \end{bmatrix} + \left[\begin{array}{c} 0 \\ \hline \Phi \end{array}\right] + \frac{1}{2\pi a\Delta x} \begin{bmatrix} \tilde{\mathtt{B}}\widehat{\mathtt{I}}_w(t) \\ \mathtt{I}_s(t) \end{bmatrix}.$$

Multiplying through by the projection matrices $\mathtt{U}$ and $\mathtt{U}^T$, we obtain the IRKA+POD reduced system

$$\partial_t \begin{bmatrix} \boldsymbol{\xi} \\ C_m\tilde{\mathtt{v}}_s \end{bmatrix} = \left[\begin{array}{c|c} \tilde{\mathtt{Q}} & \begin{pmatrix} \tilde{\mathtt{Z}}_s & 0 \end{pmatrix} \mathtt{U} \\ \hline \mathtt{U}^T \begin{pmatrix} (c/2)\tilde{\mathtt{C}} \\ 0 \end{pmatrix} & \mathtt{U}^T\mathtt{H}_s\mathtt{U} \end{array}\right] \begin{bmatrix} \boldsymbol{\xi} \\ C_m\tilde{\mathtt{v}}_s \end{bmatrix} - \begin{bmatrix} 0 \\ \mathtt{U}^T\mathtt{N}(\mathtt{U}\tilde{\mathtt{v}}_s, \mathtt{w}_s) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathtt{U}^T\overline{\Phi} \end{bmatrix} + \frac{1}{2\pi a\Delta x} \begin{bmatrix} \tilde{\mathtt{B}}\widehat{\mathtt{I}}_w(t) \\ \mathtt{U}^T\mathtt{I}_s(t) \end{bmatrix} \tag{4.17}$$

where the new "Hines" matrix is now completely dense, but of dimension $(k_w + k_s) \times (k_w + k_s)$.

The final step is to reduce the nonlinear term, which follows the same procedure as in §3.2.2. In this way we obtain a basis for the nonlinear term by taking snapshots of $\mathtt{N}$, and then using the DEIM to obtain the set of k_s spatial interpolation points $\mathbf{z}$.

We define the reduced strong gating variables by

$$\mathtt{w}_s = \mathtt{P}\tilde{\mathtt{w}}_s$$

and we approximate the strong nonlinear term $\mathtt{N}$ by applying the matrices obtained from the DEIM as in (3.14). Substituting these into (4.17) and simplifying yields the fully reduced strong-weak (RSW) system

$$\begin{aligned}\partial_t \begin{bmatrix} \boldsymbol{\xi} \\ C_m \tilde{\mathtt{v}}_s \end{bmatrix} = \left[\begin{array}{c|c} \tilde{\mathtt{Q}} & \begin{pmatrix} \tilde{\mathtt{Z}}_s & 0 \end{pmatrix} \mathtt{U} \\ \hline \mathtt{U}^T \begin{pmatrix} (c/2)\tilde{\mathtt{C}} \\ 0 \end{pmatrix} & \mathtt{U}^T \mathtt{H}_s \mathtt{U} \end{array}\right] \begin{bmatrix} \boldsymbol{\xi} \\ C_m \tilde{\mathtt{v}}_s \end{bmatrix} - \begin{bmatrix} 0 \\ \mathtt{RN}(\mathtt{U}(:, \mathbf{z})\tilde{\mathtt{v}}_s, \tilde{\mathtt{w}}_s) \end{bmatrix} \\ + \begin{bmatrix} 0 \\ \mathtt{U}^T \overline{\Phi} \end{bmatrix} + \frac{1}{2\pi a \Delta x} \begin{bmatrix} \tilde{\mathtt{B}}\widehat{\mathtt{I}}_w(t) \\ \mathtt{U}^T \mathtt{I}_s(t) \end{bmatrix}\end{aligned}$$

where $\mathrm{R} \in \mathbb{R}^{k_s \times k_s}$.

4.1.3 Results for the Reduced Strong-Weak Fiber

The uniform fiber is not morphologically interesting, but it can be used as a proof-of-concept. Since the RSW model is only really useful for weakly excitable dendrites, we apply the MIG channel model (Table B.3) to the fiber. I run 20 simulations of 1000 ms each with 1000 step current inputs, each lasting 0–2 ms and having amplitudes of 0–500 pA, applied to random locations on the fiber.

To assess the accuracy of this new method, I simulate the full system and two RSW systems, with x_T located 500 μm and 400 μm, respectively, from the distal end, and I compute the spike-capturing statistics as described in §3.3.1. I also simulate the POD+DEIM model in order to compare its accuracy to that of the RSW model. For the full and POD+DEIM system simulations, the inputs are permitted to arrive at the transition compartment, but in the RSW model, since this compartment is absent, I redirect those inputs into the adjacent compartment on the strong fiber.

Table 4.1 shows the performance results of these simulations. It is not totally clear whether or not the RSW model has improved upon the spike-capturing accuracy of the POD+DEIM model, but at least these results show that it is competitive. Note that the location of x_T has an effect both on the simulation speed and on the accuracy. Furthermore, I have used $k_w = 10$ for both RSW systems, and varying this parameter may have an effect on the accuracy as well, and it will certainly affect the speed-up.

It appears at first glance that the RSW system is much slower than the POD+DEIM system. However, profiling the code in MATLAB has shown that much extra time is spent in directing the inputs to the proper strong and weak locations in vectors, a task which is absent from the POD+DEIM code. Also, updating the "Hines" matrix and performing Gaussian Elimination seems to take a bit longer, which is counterintuitive. However, the expectation is that with a bit more optimization and perhaps some very inexpensive pre-processing, the timings can be improved to be more in line with what we would expect from the POD+DEIM simulations.

Table 4.1: Top section: Performance of reduced strong-weak model of the MIG fiber as compared with the full model. Here $k_w = 10$, $N = 1001$, and x_T is located 500 μm from the distal end. Middle section: Same as the top section, but now x_T is located 400 μm from the distal end. Bottom section:performance of the POD+DEIM model (here $k_v = k_f$) on the same fiber with the same inputs.

Reduced strong-weak model: x_T located 500 μm from distal end

k_s	Speed-up	% Matched	% Mismatched	Γ
10	3.2×	98.9	15.7	0.905
20	2.9×	98.1	2.3	0.978
25	2.7×	98.5	3.4	0.974

Reduced strong-weak model: x_T located 400 μm from distal end

k_s	Speed-up	% Matched	% Mismatched	Γ
10	3.3×	98.9	22.4	0.863
20	3.0×	97.7	3.4	0.970
25	2.7×	98.5	2.3	0.980

POD+DEIM model

k_v	Speed-up	% Matched	% Mismatched	Γ
10	5.5×	75.4	12.5	0.807
20	5.0×	95.8	3.1	0.962
25	4.6×	99.2	0.8	0.992

4.2 Generalizing the Strong-Weak Model to Arbitrary Morphologies

The RSW model can be extended to handle arbitrary branched neurons, with two main changes that must be considered. First, in partitioning the cell into strong and weak components, there is a possibility that the branches of the individual components will not be ordered to take advantage of the improvements of (Hines, 1984) that permit efficient Gaussian Elimination. Hence we must reorder the branches locally and reindex the compartments locally in order to achieve this result. Second, the

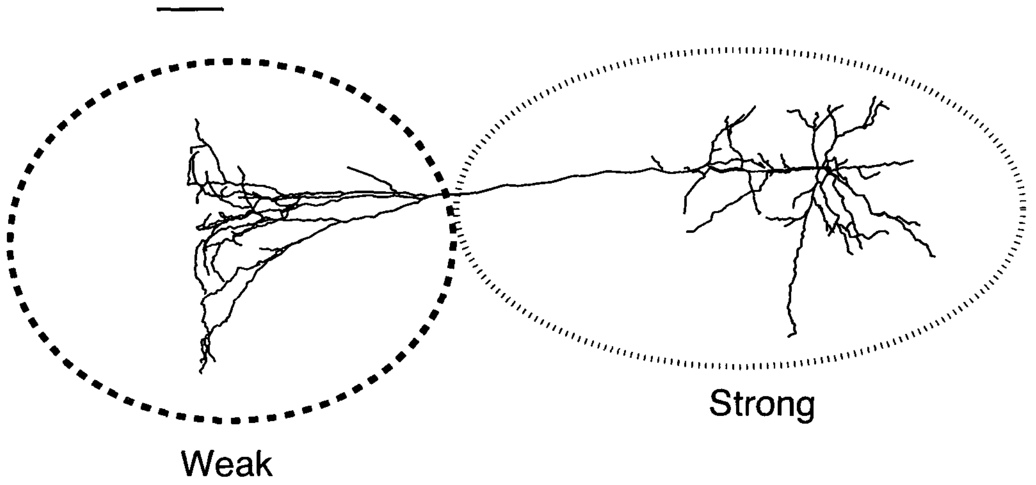


Figure 4.2: Partitioning a realistic cell into strong and weak components. This cell is p18, a pyramidal cell from the cerebral cortex (Vetter et al., 2001). The strong component encompasses dendrites that are close to the soma, where the active membrane properties will have the largest effect, while the most distal tuft comprises the weak part, since there the dendrites behave more like passive cables. The scale bar indicates 100 μm.

entries of the coupled "Hines" matrix resulting from the discretization of the transition condition will change because real neuronal morphologies will have non-uniform branch radii. Once these changes are taken care of, writing the generalized RSW model and discretizing it are straightforward.

For a general morphology, assume that we have placed the transition point x_T on branch b_T. This naturally segments the morphology into two components, one strong and one weak, as shown in Figure 4.2. The weak component consists of the segment $[0, x_T]$ of branch b_T and all branches that are its descendants. The strong component is then the complement of this set, namely the segment $[x_T, \ell_{b_T}]$ of branch b_T and all branches which are not its descendants. Hence there are a total of $\mathcal{B}_s$ and $\mathcal{B}_w$ branches in the strong and weak parts, respectively.

With this partition in place, it is necessary to reorder the branches in the strong and weak parts. This means that we assign local branch indices according to decreasing dendritic depth with respect to a specific reference point. For the strong part the soma is the reference point, and for the weak system it is the transition point x_T. This reordering allows for construction of a coupled "Hines" matrix for which Gaussian Elimination can efficiently be used. As a consequence of the reordering, the transition point x_T lies at the proximal end of branch $\beta_w = \mathcal{B}_w$ in the local ordering of the weak part, but x_T lies at the distal end of branch $\beta_s \in [1, \mathcal{B}_s]$. Thus the absolute voltages for the strong and weak parts of branch b_T in the *local* indexing are denoted by $V_{\beta_s,s}$ and $V_{\beta_w,w}$, and the radii of these components are denoted by $a_{\beta_s,s}$ and $a_{\beta_w,w}$.

The boundary conditions, initial conditions, and soma conditions for the strong-weak model are the same as those given in (2.19)–(2.26), but we must include conditions at the transition point to account for the strong-weak coupling. The voltage at the transition point x_T is given by

$$V_T(t) = V_{\beta_s,s}(x_T,t) = V_{\beta_w,w}(x_T,t), \quad v_T(t) = V_T(t) - \overline{V}_T. \tag{4.18}$$

We assume that the dendritic radii are not discontinuous in space, and thus continuity of current at the transition point requires

$$\partial_x v_{\beta_s,s}(x_T,t) = \partial_x \widehat{v}_{\beta_w,w}(x_T,t). \tag{4.19}$$

Now we discretize the neuron in space, which yields N_s and N_w compartments in the strong and weak parts, respectively, as well as the transition compartment, for a total of $N = N_s + N_w + 1$ compartments. This implies that the strong and weak voltage vectors are

$$\mathrm{v}_s \in \mathbb{R}^{N_s}, \quad \widehat{\mathrm{v}}_w \in \mathbb{R}^{N_w}.$$

To construct the coupled "Hines" matrix properly we will need the *local* indices of the compartments adjacent to the transition compartment. These local indices are denoted by N_w and N_T for the adjacent strong and weak compartments, respectively. Using these local indices, the voltage of the adjacent weak compartment is denoted by $\widehat{\mathrm{v}}_w^{N_w}$, and the voltage of the adjacent strong compartment is denoted by $\mathrm{v}_s^{N_T}$. The voltage of the transition compartment is denoted as before as v_T.

Discretizing (4.19) yields

$$\left[\frac{\mathrm{v}_s^{N_T} - \mathrm{v}_T}{\Delta x}\right] = \left[\frac{\mathrm{v}_T - \widehat{\mathrm{v}}_w^{N_w}}{\Delta x}\right],$$

where Δx is the compartment length for branch b_T. Using the continuity of potential condition of (4.18) we solve for v_T:

$$\mathrm{v}_T = \frac{1}{2}\mathrm{v}_s^{N_T} + \frac{1}{2}\widehat{\mathrm{v}}_w^{N_w}.$$

Now we are ready to apply the generalized cable equation's second-derivative operator. We can use our previously derived results (see Appendix A) for all compartments except those adjacent to the transition compartment.

Consider first the adjacent strong compartment. Discretizing at this compartment yields

$$\begin{aligned}\partial_x\big((\mathsf{a}_s^{N_T})^2\partial_x\mathsf{v}_s^{N_T}\big) &= (\mathsf{a}_s^{N_T})^2\partial_{xx}\mathsf{v}_s^{N_T} + 2\mathsf{a}_s^{N_T}\partial_x\mathsf{a}_s^{N_T}\partial_x\mathsf{v}_s^{N_T} \\ &\approx \frac{(\mathsf{a}_s^{N_T})^2}{(\Delta x)^2}\left[\mathsf{v}_s^{N_T+1} - 2\mathsf{v}_s^{N_T} + \mathsf{v}_T\right] + 2\mathsf{a}_s^{N_T}\left[\frac{\mathsf{a}_s^{N_T+1} - \mathsf{a}_T}{2\Delta x}\right]\left[\frac{\mathsf{v}_s^{N_T+1} - \mathsf{v}_T}{2\Delta x}\right],\end{aligned}$$

where a_T is the radius of the transition compartment. After substituting in (??) for v_T, followed by much tedious algebra and gathering like terms, we arrive at the following expressions for the voltage coefficients:

$$\begin{aligned}\partial_x\big((\mathsf{a}_s^{N_T})^2\partial_x\mathsf{v}_s^{N_T}\big) \approx \mathsf{v}_s^{N_T+1} &\left[\frac{(\mathsf{a}_s^{N_T})^2}{(\Delta x)^2} + \frac{\mathsf{a}_s^{N_T}}{2(\Delta x)^2}\big(\mathsf{a}_s^{N_T+1} - \mathsf{a}_T\big)\right] \\ + \mathsf{v}_s^{N_T} &\left[-\frac{3}{2}\frac{(\mathsf{a}_s^{N_T})^2}{(\Delta x)^2} + \frac{\mathsf{a}_s^{N_T}}{4(\Delta x)^2}\big(\mathsf{a}_T - \mathsf{a}_s^{N_T+1}\big)\right] \\ + \widehat{\mathsf{v}}_w^{N_w} &\left[\frac{1}{2}\frac{(\mathsf{a}_s^{N_T})^2}{(\Delta x)^2} + \frac{\mathsf{a}_s^{N_T}}{4(\Delta x)^2}\big(\mathsf{a}_T - \mathsf{a}_s^{N_T+1}\big)\right] \\ = \chi_{s,1}\mathsf{v}_s^{N_T+1} &+ \chi_{s,2}\mathsf{v}_s^{N_T} + \chi_{s,3}\widehat{\mathsf{v}}_w^{N_w},\end{aligned}$$

where $\chi_{s,1}$, $\chi_{s,2}$, and $\chi_{s,3}$ are the coefficients of the voltage terms. Similarly, for the adjacent weak compartment we find

$$\begin{aligned}
\partial_x\big((\mathsf{a}_w^{N_w})^2\partial_x\widehat{\mathsf{v}}_w^{N_w}\big) \approx{} & \mathsf{v}_s^{N_T}\left[\frac{1}{2}\frac{(\mathsf{a}_w^{N_w})^2}{(\Delta x)^2} + \frac{\mathsf{a}_w^{N_w}}{4(\Delta x)^2}\big(\mathsf{a}_T - \mathsf{a}_w^{N_w-1}\big)\right] \\
& + \widehat{\mathsf{v}}_w^{N_w}\left[-\frac{3}{2}\frac{(\mathsf{a}_w^{N_w})^2}{(\Delta x)^2} + \frac{\mathsf{a}_w^{N_w}}{4(\Delta x)^2}\big(\mathsf{a}_T - \mathsf{a}_w^{N_w-1}\big)\right] \\
& + \widehat{\mathsf{v}}_w^{N_w-1}\left[\frac{(\mathsf{a}_w^{N_w})^2}{(\Delta x)^2} + \frac{\mathsf{a}_w^{N_w}}{2(\Delta x)^2}\big(\mathsf{a}_w^{N_w-1} - \mathsf{a}_T\big)\right] \\
={} & \chi_{w,1}\mathsf{v}_s^{N_T} + \chi_{w,2}\widehat{\mathsf{v}}_w^{N_w} + \chi_{w,3}\widehat{\mathsf{v}}_w^{N_w-1}.
\end{aligned}$$

Now we are ready to write the complete coupled strong-weak system. Letting

$$c = \frac{1}{2R_a\mathsf{a}_s^{N_T}} \quad \text{and} \quad c_w = \frac{1}{2R_a\mathsf{a}_w^{N_w}}$$

and defining the coordinate vectors

$$e_{N_w} \in \mathbb{R}^{N_w(m+1)} \quad \text{and} \quad e_{N_T} \in \mathbb{R}^{N_s},$$

we can write the coupling matrices as

$$\begin{aligned}
\mathsf{Z}_w &= c_s\chi_{s,3}(e_{N_T}e_{N_w}^T) \in \mathbb{R}^{N_s \times N_w(m+1)} \\
\mathsf{Z}_s &= \frac{c_w}{C_m}\chi_{w,1}(e_{N_w}e_{N_T}^T) \in \mathbb{R}^{N_w(m+1)\times N_s}
\end{aligned}$$

and let $\mathtt{H}_s \in \mathbb{R}^{N_s\times N_s}$ be the Hines matrix for the strong part and $\mathtt{Q}$ be the quasi-active matrix for the weak part. Then the coupled strong-weak "Hines" matrix is

$$\mathtt{H} = \left[\begin{array}{c|c} \mathtt{Q} & \mathtt{Z}_s \\ \hline \mathtt{Z}_w & \mathtt{H}_s \end{array}\right]. \tag{4.20}$$

Thus the discretized strong-weak system is

$$\partial_t \begin{bmatrix} \widehat{\mathtt{v}}_w \\ \widehat{\mathtt{w}}_w \\ C_m \mathtt{v}_s \end{bmatrix} = \mathtt{H} \begin{bmatrix} \widehat{\mathtt{v}}_w \\ \widehat{\mathtt{w}}_w \\ C_m \mathtt{v}_s \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \mathtt{N}(\mathtt{v}_s(t), \mathtt{w}_s(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \overline{\Phi} \end{bmatrix} + \frac{1}{2\pi a \Delta x} \begin{bmatrix} \mathtt{B}\widehat{\mathtt{I}}_w(t) \\ \mathtt{I}_s(t) \end{bmatrix}, \tag{4.21}$$

which is of the same form as (4.14). The notable exception here is that the inputs are now given according to their local indices.

Model reduction of this system follows in the same manner as that of §4.1.1–4.1.2, yielding the reduced strong-weak system for arbitrary morphologies:

$$\begin{aligned} \partial_t \begin{bmatrix} \boldsymbol{\xi} \\ C_m \tilde{\mathtt{v}}_s \end{bmatrix} = \left[\begin{array}{c|c} \tilde{\mathtt{Q}} & \left(\tilde{\mathtt{Z}}_s\right)\mathtt{U} \\ \hline \mathtt{U}^T\left((c_s \chi_{s,3})\tilde{\mathtt{C}}\right) & \mathtt{U}^T \mathtt{H}_s \mathtt{U} \end{array}\right] \begin{bmatrix} \boldsymbol{\xi} \\ C_m \tilde{\mathtt{v}}_s \end{bmatrix} - \begin{bmatrix} 0 \\ \mathtt{RN}(\mathtt{U}(:, \mathbf{z})\tilde{\mathtt{v}}_s, \tilde{\mathtt{w}}_s) \end{bmatrix} \\ + \begin{bmatrix} 0 \\ \mathtt{U}^T \overline{\Phi} \end{bmatrix} + \frac{1}{2\pi a \Delta x} \begin{bmatrix} \tilde{\mathtt{B}}\widehat{\mathtt{I}}_w(t) \\ \mathtt{U}^T \mathtt{I}_s(t) \end{bmatrix}. \end{aligned} \tag{4.22}$$

One important point should be made concerning the selection of snapshots for both the POD and DEIM. Although it may seem reasonable to take snapshots of the strong branches from the simulation of the strong-weak system, this yielded very poor results for realistic morphologies. However, if the snapshots are taken of these branches from the full system simulation then the results improve dramatically.

4.3 Results on Branched Cells

The RSW model is in its early stages of development, but the goal of this section is not to demonstrate a complete solution. Rather, I offer evidence that this model can yield higher accuracy than the POD and DEIM reduced model, and that the RSW model may permit accurate simulation of larger, more highly branched cells than are currently feasible.

I have tested the RSW model on two branched cells with two different weakly excitable channel models. The familiar forked neuron from the previous sections appears again as the most basic branched cell to test. In order to show the utility of the model on larger cells, I use neuron p18 from (Vetter et al., 2001), which is the pyramidal cell shown in Figure 4.2. For both cells I use the MIG channel model of Table B.3, but I also use a modified version of this model to demonstrate the effect of having purely passive dendrites beyond a certain distance from the soma. This model, called MIG-P, contains the exact same parameters as the MIG model except that the conductances (except the leak conductance) are set to zero for compartments whose distance to the soma is greater than 400 μm.

4.3.1 Forked Neuron Results

Consider again the forked neuron which has one parent branch and two child branches, all with radius 1 μm and length 500 μm. The transition point x_T is located 20 μm from the distal end of the parent branch. Using a compartment length of

$\Delta x = 1$ μm, the discretized cell consists of $N = 1501$ compartments, and the strong and weak parts have $N_s = 481$ and $N_w = 1019$ compartments, respectively. I ran two different simulation sets, one with the MIG ion channel model and one with the MIG-P model. I ran 20 simulations of 1000 ms each using the full, POD+DEIM, and RSW models. For each simulation I applied 1000 step current inputs lasting 0–5 ms and having amplitudes of 0–250 pA to random locations. I computed the performance statistics for both reduced models against the full model, and these results are summarized in Table 4.2.

Note that when the MIG-P model is used, the RSW system is significantly more accurate at smaller reduced system sizes than the POD+DEIM system is. While the RSW system's timings are worse compared to the POD+DEIM system's, the RSW code is not yet optimized. Therefore I believe that, as mentioned in §4.1.3, with a better implementation the RSW system's timings should scale similarly to those of the POD+DEIM system. Also worth noting is that for the MIG-P model, the RSW system of size $k_s = 40$ achieves a $\mathbf{\Gamma}$ that is about 0.1 higher than that of the best POD+DEIM system ($k_v = 60$). Hence I hypothesize that when the weak part of the neuron consists of dendrites with mainly passive conductances, the RSW model will perform better than if there are active conductances present, even if these active ones have very little effect on the voltage dynamics.

Table 4.2: Performance of RSW and POD+DEIM models of the MIG and MIG-P fork as compared with the full model. Here $k_w = 10$, $N = 1501$, and x_T is located 20 μm from the distal end of the parent branch.

RSW model, MIG ion channel model

k_s	Speed-up	% Matched	% Mismatched	Γ
30	6.8×	98.1	4.6	0.967
40	5.7×	100	5.4	0.972

POD+DEIM model, MIG ion channel model

k_v	Speed-up	% Matched	% Mismatched	Γ
30	16.8×	97.1	2.9	0.971
40	13.4×	97.1	2.9	0.971

RSW model, MIG-P ion channel model

k_s	Speed-up	% Matched	% Mismatched	Γ
30	9.2×	96.5	14.5	0.904
40	7.5×	98.3	1.2	0.985

POD+DEIM model, MIG-P ion channel model

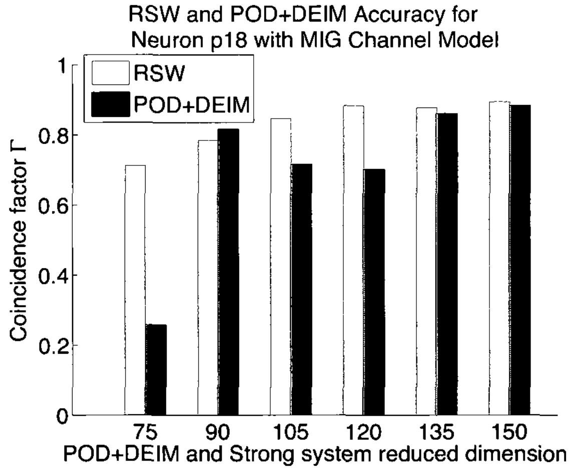
k_v	Speed-up	% Matched	% Mismatched	Γ
30	16.3×	96.5	23.3	0.850
40	13.2×	97.1	20.2	0.872
50	10.6×	97.1	19.0	0.879
60	8.4×	97.1	15.7	0.899
70	6.2×	97.1	17.8	0.887

4.3.2 Neuron p18 Results

With the success of the RSW model on the forked neuron, it is now time to examine its performance on a realistic cell. The morphologies used in §3.4.3 are not really good candidates for the RSW model because they lack very well-separated weak and strong parts. In this section I consider neuron p18, a pyramidal cell from the cerebral cortex (Vetter et al., 2001) which has two distinct tufts of dendrites connected by a long branch, as shown in Figure 4.2. This neuron is the most highly-branched one considered thus far, having 132 dendrites, and nearly half of these are contained in the weak part, indicating the potential for a large reduction in dimension.

I discretize this cell with a desired compartment length of $\Delta x = 2$ μm, yielding a model with 3996 compartments. The transition compartment is located $\sim$ 94 μm from the distal end of dendrite 60 (see Figure 4.4), leading to strong and weak dimensions of $N_s = 1898$ and $N_w = 2097$. The weak system is approximated using $k_w = 15$. Like the fork studies of the previous section, I run sets of 20 simulations of 1000 ms each using the MIG and the MIG-P ion channel models. During each simulation I apply 2000 step currents lasting 0–2 ms and having amplitudes of 0–1500 pA to random dendritic locations. In order to reduce numerical instabilities in the computed solutions, I use a time-step of $\Delta t = 0.01$.

The results of these simulations are shown in the histograms of Figure 4.3. Note that the RSW model consistently performs better than the POD+DEIM model. The RSW model is still significantly slower than the POD+DEIM one, but as mentioned before this can be improved with optimized code. Furthermore, as shown in Figure 4.4, the RSW voltage traces more closely match those of the full system. This offers evidence that the RSW model can improve accuracy for large cells using smaller reduced system sizes than the POD+DEIM model alone.



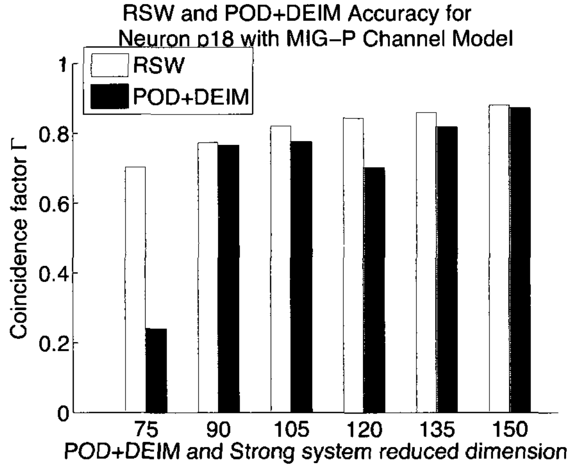


Figure 4.3: Coincidence factors **Γ** show the accuracy of the RSW and POD+DEIM models when compared to the full model for neuron p18. For the RSW system $k_w = 15$, and the values of k_s for the RSW systems and $k_v = k_f$ for the POD+DEIM systems are given on the x-axis. Top: using the MIG ion channel model. Bottom: using the MIG-P ion channel model.

4.4 Discussion

This section has provided an explicit derivation of the RSW model and yielded evidence that this model can improve upon the accuracy of the linear and nonlinear reduced models alone, but it is far from an exhaustive study. Aside from improving

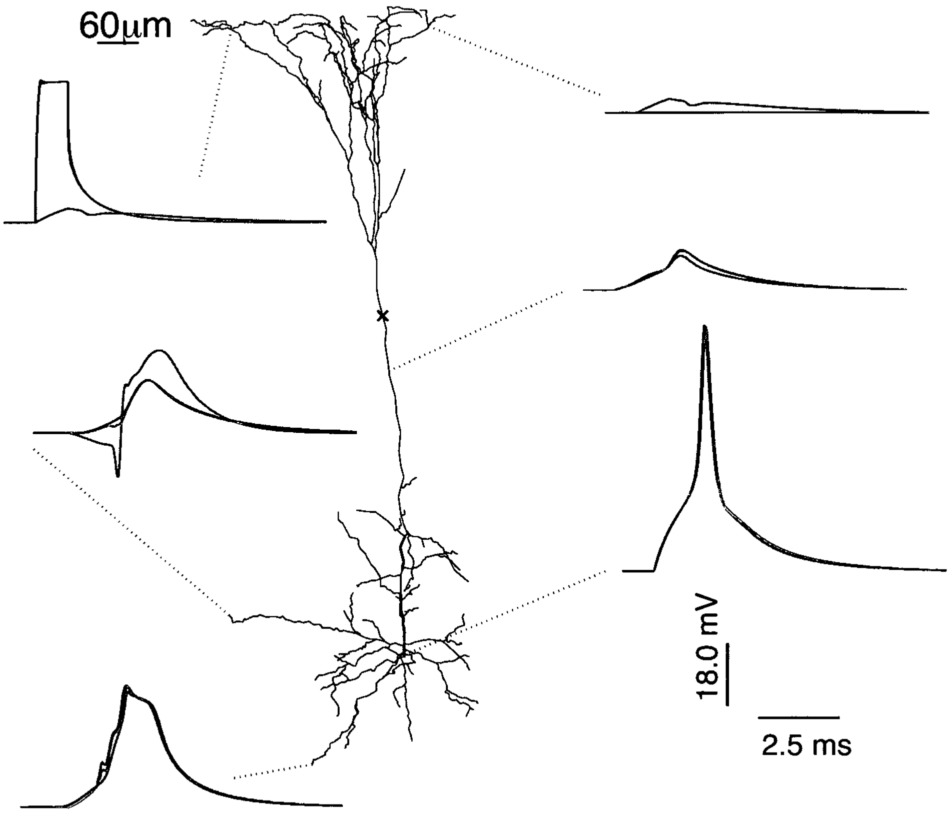


Figure 4.4: Dendritic voltage traces for cell p18 with MIG channel model. Three computed solutions are shown: the full system (blue), the RSW system (green), and the POD+DEIM system (red). The dimensions for the reduced systems are $k_v = k_f = k_s = 120$, and for the weak part of the RSW system $k_w = 15$. The transition location is indicated by the black × on the cell. Note that the voltage traces computed by the RSW system more closely match the full system solutions than the POD+DEIM solutions do, and also at locations in the weak part the POD+DEIM solutions do not match the full model with any accuracy.

the computational efficiency by optimizing the present codes, there are many considerations that will need to be addressed so that this model will be applicable to a wider class of cells, and I present the most pressing issues here.

Selection of the Transition Location

Perhaps the most critical parameter, but the least studied in this thesis, is the transition location x_T. Choosing a transition location that is too close to the soma will cut out important active dendritic segments, but choosing x_T to be too far out in the morphology may leave too many passive segments that are not well-approximated in the strong part. My method for selecting this location has been by inspection of the morphology. However, for a general-purpose implementation, this ought to be algorithmic and should remove much of the burden of selection from the user. For instance, a transition location selection algorithm could suggest candidate locations by analyzing voltage data to determine where the voltage attenuates to a small fraction of the amplitude of an action potential.

Multiple Transition Locations

Depending on the branching pattern of the morphology, it may make sense to consider having multiple transition locations. This could occur if there are multiple apical tufts, or if there is not a sufficiently long dendrite connecting the weak and strong parts. If there are p transition points, then the complete weak system is really a set of p smaller weak systems that are each coupled to the strong system but not to each other, leading to a quasi-active matrix of the block diagonal form

$$\mathsf{Q} = \begin{bmatrix} \mathsf{Q}_1 & & \\ & \ddots & \\ & & \mathsf{Q}_p \end{bmatrix}, \tag{4.23}$$

where the matrices on the diagonal correspond to the individual small weak systems.

To compute the reduction of weak systems of this form, there are two options, both of which incur extra computational cost and which alter the entries of the coupled

"Hines" matrix. The first solution is to consider one weak system with multiple observables, meaning that we use (4.23) as the linear system matrix to be reduced. In order to obtain the same accuracy as a system of the same size but with just one observable, the reduced system will need to be larger. The second solution is to consider the matrices $\mathtt{Q}_1, \ldots, \mathtt{Q}_p$ in isolation and compute p different reduced systems, each having (potentially different) dimensions k_{w_j}, $\forall j \in [1, p]$, meaning that the total reduced weak dimension is $k_w = \sum_{j=1}^{p} k_{w_j}$. It is unclear at this point which of these two approaches is superior, but it is obvious that k_w will need to be larger than it would if only one transition point is used. This degrades the performance of the RSW model because more computational effort is focused on resolving the passive system, when those extra dimensions are intuitively more valuable when used in the reduction of the strong system.

Input to the Transition Location

As noted in §4.1, the present derivation of the RSW model does not explicitly handle inputs to the transition compartment. While the simple solution is to redirect such inputs into adjacent compartments, this is not rigorous and is intellectually unsatisfying. The justification for this method is that as $\Delta x \to 0$ this will not matter, and thus such shifting of inputs should have little effect if the compartment length is small enough. Still, solutions computed in this way this can only be regarded as an approximation to the dynamics of the model given the true input pattern. Thus it is important, for accuracy and rigor, to explicitly derive how to handle inputs to the transition compartment.

Alternating Solutions of the Strong and Weak Systems

Since the RSW system consists of weak and strong systems that are coupled at the transition location, it may be possible to efficiently solve (4.22) in a two-step process. The first step is to solve for the weak state variables, using the strong voltage variables as inputs. At the next time-step, we solve for the strong state variables, using the weak voltage variables as inputs. The process is then repeated, so that we alternate solving the strong and weak systems independently. In this way the system would likely be less accurate, but would also be faster, and so perhaps this speed gain would permit using a smaller time-step, which could then recover the lost accuracy.

Chapter 5

Future Work

Each of the model reduction techniques presented in the previous chapters prompt a number of questions about improving their efficiency and functionality, as well as about how to optimize certain parameters. However, rather than enumerate many small details, I will focus on two concerns. The first is a way to include more realistic biophysics in the model and the second is a possible way to improve the performance of the models which use the POD and DEIM.

5.1 Modeling Synaptic Inputs via Receptors

Synaptic inputs can be modeled more accurately by kinetic equations than by alpha functions or similar conductance time courses (Destexhe et al., 1994). In the cell, synaptic input occurs when a presynaptic neuron releases neurotransmitter into the synaptic cleft. Neurotransmitter activates receptors on the synapse, allowing ions to flow into the cell and thus induce changes in voltage. These processes are efficiently modeled by (usually) first-order differential equations that dictate the fraction r of open receptors of a given type:

$$\frac{dr}{dt} = k_+[T](t)\big(1 - r(t)\big) - k_-r, \tag{5.1}$$

where $[T]$ is the concentration (in mM) of neurotransmitter, k_+ is the forward rate constant in $(\text{mM ms})^{-1}$, and k_- is the backward rate constant in $(\text{ms})^{-1}$ (Destexhe et al., 1998).

To include these types of inputs in the reduced models is rather straightforward, but these ODE's must be solved at each compartment. When $[T]$ has the form of a step function, (5.1) can be solved analytically (Destexhe et al., 1994), but for more general dosage functions time-stepping schemes must be used. Due to this computational cost, efficient numerical methods must be applied if the reduced systems are to yield fast simulations. One technique is to implement a shutoff mechanism like that of §3.5 so that (5.1) is only solved for compartments where r is not close to zero. However, this shutoff mechanism is likely to be more effective for fast synapses (primarily excitatory ones) than for slow synapses.

Receptors fall into two categories, excitatory and inhibitory, depending on the type of neurotransmitter which activates them. AMPA and NMDA receptors are excitatory and are activated by glutamate, while GABA$_A$ and GABA$_B$ receptors are inhibitory and are activated by GABA (Destexhe et al., 1998). AMPA and GABA$_A$ operate on much faster timescales than NMDA and GABA$_B$ do. For now, we ignore GABA$_B$ because it requires second-order kinetics. Translating the receptor kinetics into input to a branched model neuron is easy because the input terms look much like currents from ion channels. On branch b the synaptic current from receptors of type Q is given by

$$I_{bQ}(x,t) = \sum_{s=1}^{S_b} G_Q r_{bsQ}(t)\delta(x - x_{bs})\big(v_b - E_Q\big), \tag{5.2}$$

where G_Q is the maximal conductance, and E_Q is the reversal potential of that receptor. One notable exception to this form of synaptic current is the NMDA current because it is voltage-dependent, a fact that poses an extra problem for model reduction.

5.1.1 Reduction of the NMDA Magnesium Block Term

The NMDA receptor is governed by a magnesium block that is in the bound state when the voltage is close to rest. When there is sufficient depolarization, Mg^{2+} is unbound from the receptor and this permits a larger synaptic current to flow. Hence the NMDA synaptic current has the form

$$I_{b,\mathrm{NMDA}}(x,t) = \sum_{s=1}^{S_b} G_{\mathrm{NMDA}} r_{bs,\mathrm{NMDA}}(t) B_b(v_b)\delta(x - x_{bs})\big(v_b - E_{\mathrm{NMDA}}\big), \tag{5.3}$$

where

$$B_b(v_b) = \frac{1}{1 + \exp(-0.062 v_b)[Mg^{2+}]_o/3.57} \tag{5.4}$$

where $[Mg^{2+}]_o$ is the concentration of magnesium outside the cell (Destexhe et al., 1998). If the $r_{bs,\mathrm{NMDA}}(t)$ term were not there, this would be a straightforward DEIM reduction, treating $B_b(v)$ just like a gating variable. But this term is coupled to $r_{bs,\mathrm{NMDA}}(t)$, which so far cannot be reduced because it is a synaptic input that is local to each compartment. Since we cannot reduce this term, we focus our attention on extracting the $B_b(v)$ term.

After discretizing the morphology, applying the POD reduction to the NMDA synaptic input term yields

$$\mathtt{I}_{\mathrm{NMDA}}(t) = \mathtt{U}^T G_{\mathrm{NMDA}} \mathtt{r}_{\mathrm{NMDA}}(t) \mathtt{B}(\mathtt{U}\tilde{\mathtt{v}})\big(\mathtt{U}\tilde{\mathtt{v}} - E_{\mathrm{NMDA}}\big). \tag{5.5}$$

The product $\mathtt{r}(t)\mathtt{B}(\mathtt{U}\tilde{\mathtt{v}})$ is actually a *pointwise* multiplication, and this is what allows us to extract $\mathtt{B}$ and reduce it. We first approximate $\mathtt{B}$ via the DEIM as

$$\mathtt{B}(\mathtt{U}\tilde{\mathtt{v}}) \approx \mathtt{R}_B \mathtt{B}(\mathtt{U}(:, \mathbf{z}_B)\tilde{\mathtt{v}}),$$

where $\mathbf{z}_B$ are the interpolation points. We can then use pointwise multiplication to write

$$\mathtt{I}_{\mathrm{NMDA}}(t) \approx G_{\mathrm{NMDA}} \mathtt{C}_B \mathtt{B}\big(\mathtt{U}(:, \mathbf{z}_B)\tilde{\mathtt{v}}\big).\mathtt{U}^T \mathtt{r}_{\mathrm{NMDA}}(t)\big(\mathtt{U}\tilde{\mathtt{v}} - E_{\mathrm{NMDA}}\big), \tag{5.6}$$

where $\mathtt{C}_B = \mathtt{U}^T \mathtt{R}_B$.

With this derivation in place, reduced order models using receptor kinetics can now be simulated. Perhaps the most pressing issue for further study is determining how the extra cost of solving the receptor ODEs and performing the reduction of the voltage-dependent Mg^{2+} block term impacts the speed of simulation, and how the reduction of this term affects the accuracy of the system.

5.2 Augmenting the POD and DEIM Snapshot Sets

For cells with weakly excitable distal dendrites, the RSW model has shown the promise of increasing accuracy when compared to the POD+DEIM model alone, but the accuracy of both can be improved by enriching the snapshot sets. Currently the snapshot sets are generated from a single suprathreshold input, which is appropriate for ion channel models with conductances that do not strongly attenuate the voltage with distance from the soma. However, for models that yield weakly excitable dendrites, the somatic spike may nearly disappear after it has propagated sufficiently far from the soma. The result of this attenuation is that branches that are farther away are underrepresented in the snapshot set, which yields a reduced system that is less accurate than desired.

As mentioned in §3.6, a solution to this problem is to augment the snapshot set with snapshots generated by multiple inputs to some of these more distal locations. The inputs would need to be sufficiently strong to elicit the dynamics of interest, otherwise the snapshots will contain too little information to appreciably improve the accuracy of the model. The inputs should also be staggered so as to prevent overlapping dynamics that would lead to a phenomenon similar to that of phantom spiking. What is needed is a method to identify the underrepresented branches and then determine those that should receive inputs and how strong these inputs should be. While this technique has yet to be tried in any rigorous setting, preliminary results have shown that it can yield more accurate results than using just one input.

5.3 Network Simulations of Reduced Cells

A natural extension of the work in this thesis is to employ reduced cells in network simulations. Prototypical reductions can be computed for cells with different biophysical properties. For example, if a network of the hippocampus is to be modeled with regions CA3 and CA1 represented, each with 1000 cells in each region, then at least two prototypes could be created: one being a reduction of a stereotypical CA3 cell and the other of a stereotypical CA1 cell. Note that these cells can have different kinetics and morphologies, as well as different reduced system sizes. However, within each brain region there are multiple (but a finite number of) subtypes of cells, and some are excitatory while others are inhibitory (Shepherd and Koch, 1998) (Traub and Miles, 1991). Hence the network population of 1000 cells may be composed of, say, 20 different prototype cells, each represented according to their proportion of occurrence in the brain. These reduced cells are then building blocks that can be linked together to form networks by implementing a synaptic transmission mechanism. This opens the door to investigation of other neuronal functions, such as synaptic plasticity, using realistic morphologies but without the price of slow simulations.

Preliminary codes have been developed as a proof of concept for very small networks (less than 20 cells), and the results are promising. In fact, if the prototype cells simulate 10 times faster than their full system counterparts, then the network simulations also show the same speed-up. Of course, the network simulation speed will be limited by the simulation speed of the slowest prototype cell. However, the drastic speed-up without sacrificing the input-output map of the individual cells means that this reduce-then-connect framework for network model reduction could be revolutionary.

Chapter 6

Conclusion

Using the tools in this thesis, models of morphologically accurate spiking cells can be reduced to systems with significantly fewer variables while maintaining the input-output properties of the original model. Balanced Truncation (BT) offers a rigorous theoretical approach to linear model reduction in order to reproduce subthreshold voltage dynamics, but it is slow and may be computationally intractable for some systems. An Iterative Rational Krylov Algorithm (IRKA) achieves the same goal, but it computes the reduced system drastically faster and also makes simulations of realistic morphologies tractable. Depending on the discretization and the morphology, the dimension of the system can be reduced by a factor of more than 10000, and the simulation time can decrease by a factor of nearly 100.

The spiking regime requires a different approach in order to capture the highly nonlinear voltage response. The Proper Orthogonal Decomposition (POD) reduces the dimension of the state variables, while a Discrete Empirical Interpolation Method (DEIM) reduces the complexity of the nonlinear terms that account for ion channel kinetics. In order for these methods to be accurate for branched cells, I have developed a branchwise orthogonalization algorithm that improves the accuracy of the POD+DEIM system, and a snapshot elimination algorithm that removes snapshots that provide little information. Using these techniques together, spike times can be accurately computed to within 2 ms with coincidence factors $\Gamma \geq 0.9$ for a broad class of realistic neurons, and simulation times can decrease by a factor of nearly 10.

While these model reduction techniques perform very well in isolation, together they can be even more effective. The Reduced Strong-Weak (RSW) model incorporates a reduced linear system to model regions with weak (linear) dynamics while using a POD+DEIM system to model the dynamics of strong (nonlinear) regions. This method is still under development, so the main contribution of this thesis is to present the theory and derivation for this model. However, I have also offered computational evidence that the RSW model can be more accurate than the linear or nonlinear reduced models individually.

Until now, model reduction of realistic spiking neurons was not robust to inputs nor faithful to the spatial description of the cell. In this thesis I have developed and implemented multiple techniques that satisfy these two requirements and outperform all current model reduction efforts on such cells. These results demonstrate that hidden within the complicated dendritic structure there exists a low-dimensional subspace that describes the neuronal dynamics. This knowledge is important not only because neuronal simulations can be drastically accelerated, but also because it encourages exploration into model reduction for other complex systems in the brain.

Appendix A

Constructing the Hines Matrix

A.1 Current Implementation

Applying finite difference schemes to the spatial derivatives involved in the cable equation and the boundary conditions is a straightforward procedure, but care must be taken to properly account for coupling at junctions in order to maintain an accurate numerical solution.

The Hines ordering implies that compartments are labeled in increasing order from most distal to most proximal on a branch, starting with the most proximal branch and ending with the soma. The latter is measured by called a branch's *depth*, which is defined as the number of branches it must traverse to reach the soma, including itself. Hence the soma compartment has depth $= 0$, roots have depth $= 1$, and so on. Without loss of generality, assume further that each branch has N compartments numbered 1 through N. This gives 6 possible compartment types whose entries in the Hines matrix we must compute:

1. Interior compartment

2. Leaf compartment: the most distal compartment of a leaf

3. Parent compartment (non-soma): compartment N of a child branch whose parent is another branch

4. Parent compartment (soma): compartment N of a root branch

5. Branch compartment: the first compartment of a parent branch

6. Soma

For clarity in the following derivations I will drop the branch subscript b and instead will indicate quantities by their compartment index k. Thus here a_k should be interpreted as $a(x_k)$, and similarly for v_k.

1: Interior compartment

From the main cable equation we compute the partial derivative term as

$$\partial_x(a^2(x)\partial_x v) = a^2(x)v_{xx} + 2a(x)a_x v_x.$$

Using the standard discretization scheme for the second partial derivative and centered differences for the first partial derivatives we obtain

$$\begin{aligned}\partial_x(a^2(x)\partial_x v) &\approx a^2(x_k)\left[\frac{v(x_{k+1})-2v(x_k)+v(x_{k-1})}{h^2}\right]\\ &+2a(x_k)\left[\frac{a(x_{k+1})-a(x_{k-1})}{2h}\right]\left[\frac{v(x_{k+1})-v(x_{k-1})}{2h}\right]\\ &=\frac{a_k^2}{h^2}\left[v_{k+1}-2v_k+v_{k-1}\right]+\frac{a_k}{2h^2}\left[a_{k+1}v_{k+1}-a_{k+1}v_{k-1}-a_{k-1}v_{k+1}+a_{k-1}v_{k-1}\right]\\ &=v_{k+1}\left[\frac{a_k^2}{h^2}+\frac{a_k}{2h^2}(a_{k+1}-a_{k-1})\right]\\ &+v_k\left[-\frac{2a_k^2}{h^2}\right]\\ &+v_{k-1}\left[\frac{a_k^2}{h^2}-\frac{a_k}{2h^2}(a_{k+1}-a_{k-1})\right].\end{aligned}$$

2: Leaf compartment

Leaf compartments satisfy $\partial_x v(\ell,t)=0$, which is discretized as

$$\partial_x v(\ell,t)\approx\frac{v_2-v_1}{h}.$$

However, because the membrane current must be accounted for from the active dynamics in the cable equation, we must divide through by one more h, bringing the final expression to

$$0=\left[\frac{1}{h^2}\right]v_2+\left[\frac{-1}{h^2}\right]v_1.$$

3: Parent compartment (non-soma)

A non-soma parent compartment is that on a child branch immediately before the branch point, i.e., it is compartment N on that branch. Such compartments follow the same derivation as interior ones, except that the v_{k+1} and a_{k+1} terms are replaced by $v_{\text{parent},1}$ and a_N, yielding

$$\begin{aligned}\partial_x(a^2(x)\partial_x v) \approx v_{\text{parent},1} &\left[\frac{a_k^2}{h^2} + \frac{a_k}{2h^2}(a_N - a_{k-1})\right] \\ + v_k &\left[-\frac{2a_k^2}{h^2}\right] \\ + v_{k-1} &\left[\frac{a_k^2}{h^2} - \frac{a_k}{2h^2}(a_N - a_{k-1})\right].\end{aligned}$$

4: Parent compartment (soma)

A non-soma parent compartment is that on a child branch immediately before the soma. Hence they follow the same derivation as interior compartments, except that the v_{k+1} and a_{k+1} terms are replaced by v_{soma} and a_N, yielding

$$\begin{aligned}\partial_x(a^2(x)\partial_x v) \approx v_{\text{soma}} &\left[\frac{a_k^2}{h^2} + \frac{a_k}{2h^2}(a_N - a_{k-1})\right] \\ + v_k &\left[-\frac{2a_k^2}{h^2}\right] \\ + v_{k-1} &\left[\frac{a_k^2}{h^2} - \frac{a_k}{2h^2}(a_N - a_{k-1})\right].\end{aligned}$$

5: Branch compartment

A branch compartment is the first one on the parent branch. These nodes satisfy the boundary conditions

$$\begin{aligned} v_{p_k}(\ell_{p_k}, t) &= v_{c_k^1}(0, t) + v_{c_k^2}(0, t) \\ a_{p_k}^2 \partial_x v_{p_k}(\ell_{p_k}, t) &= a_{c_k^1}^2 \partial_x v_{c_k^1}(0, t) + a_{c_k^2}^2 \partial_x v_{c_k^2}(0, t). \end{aligned}$$

I discretize the second of these conditions by forward and backward differences using the radii that are one compartment *away* from the branch point to explicitly show the effect of varying radii entering the branch; in the limit as $h \to 0$ these are the same values:

$$a_{p,2}^2 \left[\frac{v_{p,2} - v_{p,1}}{h_p} \right] = a_{c_1,N}^2 \left[\frac{v_{p,1} - v_{c_1,N}}{h_{c_1}} \right] + a_{c_2,N}^2 \left[\frac{v_{p,1} - v_{c_2,N}}{h_{c_2}} \right].$$

Gathering like terms to the left and scaling by $1/h_p$ because of the membrane current yields

$$\begin{aligned} 0 = v_{p,1} &\left[-\frac{a_{p,2}^2}{h_p^2} - \frac{a_{c_1,N}^2}{h_{c_1} h_p} - \frac{a_{c_2,N}^2}{h_{c_2} h_p} \right] \\ &+ v_{p,2} \left[\frac{a_{p,1}^2}{h_p^2} \right] \\ &+ v_{c_1,N} \left[\frac{a_{c_1,N}^2}{h_{c_1} h_p} \right] \\ &+ v_{c_2,N} \left[\frac{a_{c_2,N}^2}{h_{c_2} h_p} \right]. \end{aligned}$$

6: Soma

The soma satisfies the three conditions (2.21) and (2.22), and hence for the dth root we write

$$\begin{aligned}\frac{\pi}{R_a A_\sigma} a_d^2(0)\partial_x v_d(0,t) \approx\ & v_{\text{soma}}\left[-\frac{\pi}{R_a A_\sigma h_d} a_{d,N}^2\right] \\ & + v_{d,N}\left[\frac{\pi}{R_a A_\sigma h_d} a_{d,N}^2\right].\end{aligned}$$

A.2 Second-order Approximations of Boundary Conditions

The derivations in the previous section describe the implementation of the codes as used to generate the results in this thesis, but they involve mixed orders of accuracy. The interior nodes are computed to second-order accuracy, but the boundary nodes (and the junction nodes) are only computed using first-order accurate schemes.

Second-order accuracy for the boundary conditions can be obtained by implementing an appropriate one-sided finite difference scheme (Niebur and Niebur, 1991). For example, the sealed end condition $\partial_x v(\ell, t) = 0$ can be discretized as

$$\partial_x v(\ell, t) \approx \frac{-3v_1 + 4v_2 - v_3}{2h},$$

and similarly this may be applied to the junction conditions to recover second-order accuracy across the whole spatial grid.

Appendix B

Ion Channel Kinetics

The following tables contain all the information pertaining to the ion channels and gating variable kinetics used in this thesis. Table B.1 is for the uniform channel model, which uses the Hodgkin-Huxley squid giant axon parameters. Table B.2 is for the non-uniform channel model, whose non-uniformity comes from an A-type K^+ current following Connor-Stevens kinetics and consistent with the graded channel distribution of (Hoffman et al., 1997). In order to simulate cells with weakly excitable dendrites, the model in Table B.3 is based on that of (Migliore et al., 1999) for I_{Na}, $I_{K_{\mathrm{DR}}}$, $I_{K_{\mathrm{A}}}$, and I_{leak}, though there have been some modifications to the time constants for l_{prox} and l_{dist}, which was done to obtain a good fit to the original functions without the need for defining them piecewise.

Table B.1: Uniform channel model and kinetics, which corresponds to the Hodgkin-Huxley squid giant axon parameters at 6.3°C (Hodgkin and Huxley, 1952).

Ionic Current	Gating Variables	$G(x)$ (mS/cm^2)	E (mV)
Leak		0.3	−54.3
I_{Na}	m^3, h	120	56
I_K	n^4	36	−77

Gating Var.	α	β	τ	w_∞
m	$\frac{0.1(v+40)}{1-\exp(-0.1(v+40))}$	$4\exp(-(v+65)/18)$	$\frac{1}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$
h	$0.07\exp(-(v+65)/20)$	$\frac{1}{1+\exp(0.1(-35-v))}$	$\frac{1}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$
n	$\frac{0.01(v+55)}{1-\exp(-0.1(v+55))}$	$\frac{1}{8}\exp(-(v+65)/80)$	$\frac{1}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$

Table B.2: Non-uniform channel model and kinetics. I_{Na} and I_K use Hodgkin-Huxley type kinetics, while I_A uses Connor-Stevens type kinetics and a spatial distribution of conductance based on Hoffman's work. For $G(x)$, x is measured in μm from the soma. Note that the I_A channel kinetics do not have α and β functions explicitly defined (Connor and Stevens, 1971) (Hoffman et al., 1997).

Ionic Current	Gating Variables	$G(x)$ (mS/cm^2)	E (mV)
Leak		0.3	−47
I_{Na}	m^3, h	60	55
I_K	n^4	20	−72
I_A	a^3, b	$\frac{47.7}{2.3}(0.2+0.002x)$	−75

Gating Var.	α	β	τ	w_∞
m	$\frac{0.38(v+29.7)}{1-\exp(-(v+29.7)/10)}$	$15.2\exp(-0.0556(v+54.7))$	$\frac{1}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$
h	$0.266\exp(-0.05(v+48))$	$\frac{3.8}{1+\exp(-(v+18)/10)}$	$\frac{1}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$
n	$\frac{0.02(v+45.7)}{1-\exp(-(v+45.7)/10)}$	$0.25\exp(-0.0125(v+55.7))$	$\frac{1}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$

Gating Var.	τ	w_∞
a	$0.3632+\frac{1.158}{1+\exp(0.0497(v+55.96))}$	$\left(\frac{0.0761\exp(0.0314(v+94.22))}{1+\exp(0.0346(v+1.17))}\right)^{1/3}$
b	$1.24+\frac{2.678}{1+\exp(0.0624(v+50))}$	$\left(\frac{1}{1+\exp(0.0688(v+53.3))}\right)^{4}$

Table B.3: Channel model and kinetics that allow weakly excitable dendrites. For $G(x)$, x is measured in μm from the soma. The model for I_{K_A} is divided into two components, proximal and distal, which are denoted $I_{K_{\text{A(prox)}}}$ and $I_{K_{\text{A(dist)}}}$, respectively. Some of the gating variables in this model have time constants τ that depend on temperature T via a so-called Q_{10} factor. The temperature is set to $T = 35°\text{C}$ to match the value used in (Migliore et al., 1999).

Ionic Current	Gating Variables	$G(x)$ (mS/cm^2)	E (mV)
I_leak		0.3	-65
I_{Na}	m^3, h, i	32	55
I_{K_DR}	n_DR	10	-90
$I_{K_{\text{A(prox)}}}$	n_prox, l_prox	$\begin{cases} 48(1+x/100) & x \leq 100\ \mu\text{m} \\ 0 & x > 100\ \mu\text{m} \end{cases}$	-90
$I_{K_{\text{A(dist)}}}$	n_dist, l_dist	$\begin{cases} 0 & x \leq 100\ \mu\text{m} \\ 48(1+x/100) & x > 100\ \mu\text{m} \end{cases}$	-90

Gating Var.	α	β
m	$\frac{0.4(v+30)}{1-\exp(-(v+30)/7.2)}$	$\frac{0.124(v+30)}{\exp((v+30)/7.2)-1}$
h	$\frac{0.03(v+45)}{1-\exp(-(v+45)/1.5)}$	$\frac{0.01(v+45)}{\exp((v+45)/1.5)-1}$
i	$\exp(0.45(v+60))$	$\exp(0.09(v+60))$
n_DR	$\exp(-0.11(v-13))$	$\exp(-0.08(v-13))$
n_prox	$\exp(-0.038(1.5+\frac{1}{1+\exp(v+40/5)}))(v-11)$	$\exp(-0.038(0.825+\frac{1}{1+\exp(v+40/5)}))(v-11)$
l_prox	$\exp(0.11(v+56))$	
n_dist	$\exp(-0.038(1.8+\frac{1}{1+\exp(v+40/5)}))(v+1)$	$\exp(-0.038(0.7+\frac{1}{1+\exp(v+40/5)}))(v+1)$
l_dist	$\exp(0.11(v+56))$	

Gating Var.	τ	w_∞	Q_{10} factor
m	$\frac{0.5}{\alpha+\beta}$	$\frac{\alpha}{\alpha+\beta}$	$2^{(T-24)/10}$
h	$\frac{0.5}{\alpha+\beta}$	$\frac{1}{1+\exp((v+50)/4)}$	$2^{(T-24)/10}$
i	$\frac{3\times10^4\beta}{1+\alpha}$	$\frac{1+0.5\exp((v+58)/2)}{1+\exp((v+58)/2)}$	
n_DR	$\frac{50\beta}{1+\alpha}$	$\frac{1}{1+\alpha}$	
n_prox	$\frac{4\beta}{1+\alpha}$	$\frac{1}{1+\alpha}$	$5^{(T-24)/10}$
l_prox	$0.2+\frac{27}{1+\exp(0.2-v/22)}$	$\frac{1}{1+\alpha}$	
n_dist	$\frac{2\beta}{1+\alpha}$	$\frac{1}{1+\alpha}$	$5^{(T-24)/10}$
l_dist	$0.2+\frac{27}{1+\exp(0.2-v/22)}$	$\frac{1}{1+\alpha}$	

Bibliography

L. F. Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50:303–304, 1999.

O. J. Ahmed and M. R. Mehta. The hippocampal rate code: anatomy, physiology and theory. *Trends in Neurosciences*, 32(6):329–338, 2009.

A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control)*. SIAM, 2005.

A. C. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: An overview. *International J. of Applied Math and Computer Science*, 11(5):1093–1121, 2001.

G. A. Ascoli. Mobilizing the base of neuroscience data: the case of neuronal morphologies. *Nature Reviews Neuroscience*, 7:318–324, 2006.

M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris*, 339:667–672, 2004.

U. S. Bhalla, D. H. Bilitch, and J. M. Bower. Rallpacks: a set of benchmarks for neuronal simulators. *Trends in Neuroscience*, 15(11):453–458, 1992.

R. Brette and W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiology*, 94:3637–3642, 2005.

N. Brunel and X.-J. Wang. What determines the frequency of fast network oscillations with irregular neural discharges? I. Synaptic dynamics and excitation-inhibition balance. *J. Neurophysiol.*, 90:415–430, 2003.

S. Chaturantabut and D. C. Sorensen. Discrete empirical interpolation for nonlinear model reduction. Technical Report TR09-05, Department of Computational and Applied Mathematics, Rice University, March 2009.

R. A. Chitwood, A. Hubbard, and D. B. Jaffe. Passive electrotonic properties of rat hippocampal CA3 interneurones. *J. Physiol.*, 515:743–56, 1999.

C. M. Colbert and E. Pan. Ion channel properties underlying axonal action potential initiation in pyramidal neurons. *Nat. Neurosci.*, 5:533–538, 2002.

J. A. Connor and C. F. Stevens. Inward and delayed outward membrane currents in isolated neural somata under voltage clamp. *J. Physiol.*, 213:1–19, 1971.

A. Destexhe, Z. F. Mainen, and T. J. Sejnowski. An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, 6:14–18, 1994.

A. Destexhe, Z. F. Mainen, and T. J. Sejnowski. Kinetic models of synaptic transmission. In C. Koch and I. Segev, editors, *Methods in Neuronal Modeling: From Ions to Networks*, pages 1–26. The MIT Press, 2nd edition, 1998.

B. Ermentrout. Reduction of conductance-based models with slow synapses to neural nets. *Neural Computation*, 6:679–695, 1994.

N. Fourcaud-Trocmé, D. Hansel, C. van Vreeswijk, and N. Brunel. How spike generation mechanisms determine the neuronal response to fluctuating inputs. *J. Neurosci.*, 23:11628–11640, 2003.

R. W. Freund. Structure-preserving model order reduction of rcl circuit equations. In W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13, pages 49–73. Springer Berlin Heidelberg, 2008.

S. C. Furtak, J. R. M. Jr., and T. H. Brown. Morphology and ontogeny of rat perirhinal cortical neurons. *J. Comp. Neurol.*, 505(5):493–510, December 2007.

K. Glover. All optimal hankel-norm approximations of linear multivariable systems and their L^{∞}-error bounds. *Int. J. Contr.*, 39:1115–1193, 1984.

N. L. Golding, W. L. Kath, and N. Spruston. Dichotomy of action-potential backpropagation in CA1 pyramidal neuron dendrites. *J. Neurophysiol.*, 86:2998–3010, 2001.

N. L. Golding, T. J. Mickus, Y. Katz, W. L. Kath, and N. Spruston. Factors mediating powerful voltage attentuation along CA1 pyramidal neuron dendrites. *J. Physiol.*, 568:69–82, 2005.

S. Gugercin, A. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM J. on Matrix Analysis and Applications*, 30:609–638, 2008.

M. E. Hasselmo, L. M. Giocomo, and E. A. Zilli. Grid cell firing may arise from interference of theta frequency membrane potential oscillations in single neurons. *Hippocampus*, 17:1252–1271, 2007.

M. Häusser. Synaptic function: Dendritic democracy. *Current Biology*, 11:R10–R12, 2001.

M. Hines. Efficient computation of branched nerve equations. *International J. of Bio-Medical Computing*, 15:69–76, 1984.

M. L. Hines and N. T. Carnevale. Neuron: a tool for neuroscientists. *The Neuroscientist*, 7:123–135, 2001.

M. L. Hines, T. Morse, M. Migliore, N. T. Carnevale, and G. M. Shepherd. Modeldb: A database to support computational neuroscience. *J. Comput. Neurosci.*, 17:7–11, 2004.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117:500–544, 1952.

D. A. Hoffman, J. C. Magee, C. M. Colbert, and D. Johnston. K^+ channel regulation of signal propagation in dendrites of hippocampal pyramidal neurons. *Nature*, 387: 869–875, 1997.

http://NeuroMorpho.org. The neuromorpho.org inventory. Accessed March 11, 2008. `http://NeuroMorpho.org`.

http://senselab.med.yale.edu/modeldb. Senselab modeldb. Accessed March 11, 2008. `http://senselab.med.yale.edu/modeldb`.

B. Hutcheon and Y. Yarom. Resonance, oscillation and the intrinsic frequency preferences of neurons. *Trends Neurosci.*, 23:216–222, 2000.

D. Johnston and D. G. Amaral. Hippocampus. In G. M. Shepherd, editor, *The Synaptic Organization of the Brain*, chapter 11, pages 417–458. Oxford University Press, 4th edition, 1998.

R. Jolivet, A. Rauch, H.-R. Lüscher, and W. Gerstner. Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *J. Comput. Neurosci.*, 21:35–49, 2006.

T. Kailath. *Linear Systems*. Prentice Hall, 1980.

E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, 4th edition, 2000.

A. R. Kellems. Efficient and accurate simulation of integrate-and-fire neuronal networks in the hippocampus. Master's thesis, Rice University, 2007.

A. R. Kellems, D. Roos, N. Xiao, and S. J. Cox. Low-dimensional, morphologically accurate models of subthreshold membrane potential. *J. Comput. Neurosci.*, 27(2): 161–176, 2009.

A. R. Kellems, S. Chaturantabut, D. C. Sorensen, and S. J. Cox. Morphologically accurate reduced order modeling of spiking neurons. *J. Comput. Neurosci.*, Published online, 2010.

T. B. Kepler, L. Abbott, and E. Marder. Reduction of conductance-based neuron models. *Biological Cybernetics*, 66:381–387, 1992.

W. M. Kistler, W. Gerstner, and J. L. van Hemmen. Reduction of the hodgkin-huxley equations to a single-variable threshold model. *Neural Computation*, 9:1015–1045, 1997.

C. Koch. *Biophysics of Computation.* Oxford University Press, Inc., 1999.

M. H. P. Kole, S. U. Ilschner, B. M. Kampa, S. R. Williams, P. C. Ruben, and G. J. Stuart. Action potential generation requires a high sodium channel density in the axon initial segment. *Nat. Neurosci.*, 11:178–186, 2008.

K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2002.

A. Lansner. Associative memory models: from the cell-assembly theory to biophysically detailed cortex simulations. *Trends in Neurosciences*, 32(3):178–186, 2009.

Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper orthogonal decomposition and its applications—part I: Theory. *J. Sound and Vibration*, 252:527–544, 2002.

B. E. Losavio, Y. Liang, A. Santamaría-Pang, I. A. Kakadiaris, C. M. Colbert, and P. Saggau. Live neuron morphology automatically reconstructed from multiphoton an confocal imaging data. *J. Neurophysiol.*, 100:2422–2429, 2008.

J. C. Magee and E. P. Cook. Somatic EPSP amplitude is independent of synapse location in hippocampal pyramidal neurons. *Nat. Neurosci.*, 3:895–903, 2000.

Z. F. Mainen and T. J. Sejnowski. Modeling active dendritic processes in pyramidal neurons. In C. Koch and I. Segev, editors, *Methods in Neuronal Modeling: From Ions to Networks*, pages 171–210. The MIT Press, 2nd edition, 1998.

H. Markram. The Blue Brain Project. *Nature Rev. Neuroscience*, 7:153–160, 2006.

J. O. Martinez. Rice-Baylor archive of neuronal morphology. Accessed May 1, 2008. `http://www.caam.rice.edu/~cox/neuromart`.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

M. Migliore, D. A. Hoffman, J. C. Magee, and D. Johnston. Role of an A-type K^+ conductance in the back-progagation of action potentials in the dendrites of hippocampal pyramidal neurons. *J. Comput. Neurosci.*, 7:5–15, 1999.

M. Migliore, C. Cannia, W. W. Lytton, H. Markram, and M. L. Hines. Parallel network simulations with neuron. *J. Comput. Neurosci.*, 21:119–129, 2006.

B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26: 17–32, 1981.

D. Morel and W. B. Levy. The cost of linearization. *J. Comput. Neurosci.*, 2009.

N. C. Nguyen, A. T. Patera, and J. Peraire. A 'best points' interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Meth. Engng.*, 73:521–543, 2008.

E. Niebur and D. Niebur. Numerical implementation of sealed-end boundary conditions in cable theory. *IEEE Transactions on Biomedical Engineering*, 38(12): 1266–1271, 1991.

P. F. Pinsky and J. Rinzel. Intrinsic and network rhythmogenesis in a reduced Traub model for CA3 neurons. *J. Comput. Neurosci.*, 1:39–60, 1994.

E. Puil, B. Gimbarzevsky, and R. M. Miura. Quantification of membrane properties of trigeminal root ganglion neurons in guinea pigs. *J. Neurophysiol.*, 55:995–1016, 1986.

G. Pyapali, A. Sik, M. Penttonen, G. Buzsaki, and D. Turner. Dendritic properties of hippocampal ca1 pyramidal neurons in the rat: Intracellular staining in vivo and in vitro. *J. of Comparative Neurology*, 391:335–352, 1998.

W. Rall. Branching dendritic trees and motoneuron membrane resistivity. *Experimental Neurology*, 1:491–527, 1959.

W. Rall. Theoretical significance of dendritic trees for neuronal input-output relations. In R. F. Reiss, editor, *Neural Theory and Modeling*. Palo Alto: Stanford University Press, 1964.

M. J. Rempe, N. Spruston, W. L. Kath, and D. L. Chopp. Compartmental neural simulations with spatial adaptivity. *J. Comput. Neurosci.*, 25(3):465–480, 2008.

L. L. Rihn and B. J. Claiborne. Dendritic growth and regression in rat dentate granule cells during late postnatal development. *Brain Res. Dev. Brain Res.*, 54(1):115–24, 1990.

M. Rudolph and A. Destexhe. Analytical integrate-and-fire neuron models with conductance-based dynamics for event-driven simulation strategies. *Neural Computation*, 18:2146–2210, 2006.

J. Sabino. *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method.* PhD thesis, Rice University, 2006.

G. M. Shepherd and C. Koch. Introduction to synaptic circuits. In G. M. Shepherd, editor, *The Synaptic Organization of the Brain*, chapter 1, pages 1–36. Oxford University Press, 4th edition, 1998.

D. Shin, D. R. Yang, and J. Choi. On the use of pseudo-spectral method in model reduction and simulation of active dendrites. *Comp. Bio. Med.*, 39:340–345, 2009.

O. Shriki, D. Hansel, and H. Sompolinsky. Rate models for conductance-based cortical neuronal networks. *Neural Computation*, 15:1809–1841, 2003.

M. E. Sorensen and S. P. DeWeerth. An algorithmic method for reducing conductance-based neuron models. *Biol. Cybern.*, 95:185–192, 2006.

R. A. Stefanescu and V. K. Jirsa. A low dimensional description of globally coupled heterogeneous neural networks of excitatory and inhibitory neurons. *PLoS Comput. Bio.*, 4(11), 2008.

G. Stuart, N. Spruston, and M. Häusser, editors. *Dendrites.* Oxford University Press, 2nd edition, 2008.

A. M. Thomson and C. Lamy. Functional maps of neocortical local circuitry. *Frontiers in Neuroscience*, 1:19–42, 2007.

Y. Timofeeva, S. J. Cox, S. Coombes, and K. Josić. Democratization in a passive dendritic tree: an analytical investigation. *J. Comput. Neurosci.*, 25:228–244, 2008.

C. B. Toris, J. L. Eiesland, and R. F. Miller. Morphology of ganglion cells in the neotenous tiger salamander retina. *J. Comp. Neurol.*, 352(4):535–59, 1995.

R. D. Traub and R. Miles. *Neuronal Networks of the Hippocampus.* Cambridge University Press, 1991.

L. N. Trefethen. Is Gauss quadrature better than Clenshaw–Curtis? *SIAM Review*, 2007.

L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, 1997.

D. Ulrich. Dendritic resonance in rat neocortical pyramidal cells. *J. Neurophysiol.*, 87:2753–2759, 2002.

P. Vetter, A. Roth, and M. Häusser. Propagation of action potentials in dendrites depends on dedritic morphology. *J. Neurophysiol.*, 85(2):926–937, 2001.

H. R. Wilson. *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience.* Oxford University Press, 1999.

M. P. Witter and E. I. Moser. Spatial representation and the architecture of the entorhinal cortex. *TRENDS in Neuroscience*, 29:671–678, 2006.

B. Woo and J. Choi. Reduced model and simulation of myelinated axon using eigenfunction expansion and singular perturbation. *Comp. Bio. Med.*, 37:1148–1154, 2007.

B. Woo, D. Shin, D. R. Yang, and J. Choi. Reduced model and simulation of neuron with passive dendritic cable: An eigenfunction expansion approach. *J. Comput. Neurosci.*, 19:379–397, 2005.

D. Yoganarasimha, X. Yu, and J. J. Knierim. Head direction cell representations maintain internal coherence during conflicting proximal and distal cue rotations: Comparison with hippocampal place cells. *J. Neurosci.*, 26(2):622–631, January 2006.