AUTHOR: **Quentin Caudron** DEGREE: **Ph.D.**

TITLE: **Neuronal Computation on Complex Dendritic Morphologies**

Neuronal Computation on Complex Dendritic Morphologies

by

Quentin Caudron

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Centre for Complexity Science

December 2012

THE UNIVERSITY OF
WARWICK

Contents

Acknowledgements

During my four years at the Centre for Complexity Science at Warwick, I have learned a great deal about mathematics, science, and research in general. In the process, established academics would become colleagues, and colleagues would become friends. Without their aid, support, and encouragement, I would simply not be where I am today.

First amongst these is my supervisor, Dr. Yulia Timofeeva, who always guided me with patience and knowledge, while letting me explore the landscape in my own way. Her expertise and advice have led me to produce work of which I am genuinely proud, and for this, I am extremely grateful.

To the talented graph theorist and mathematician, Simon Donnelly, I offer my deepest gratitude. Discussions with you would always bear fruit, the end result of which is evident in this thesis.

I wish to thank the extremely talented mathematicians, Jamie Harris and Sam Brand, for many discussions spent poring over blackboards.

My family remain a neverending source of support and motivation. To my wife, Madi, for her love and for my sanity; to my father, for keeping things in perspective; to my mother, for always being there; and to my siblings, for being yourselves; thank you.

Declarations

The work in this thesis is a presentation of my original research. Every effort has been made to credit the authors of the peer-reviewed literature which provide the foundations upon which this work is built, and the collaborators who contributed their skill and time in the development of this work.

Parts of the work in Chapter 4 were done in collaboration with Simon Donnelly, of Edinburgh's Neuroinformatics Doctoral Training Centre, whose help was fundamental to my understanding of aspects of graph theory, and subsequently to my development of these algorithms; and Sam Brand, of Warwick Complexity, who introduced me to the Feynman-Kac relation between some differential equations and stochastic processes. This work has been published in the Journal of Mathematical Neuroscience, as

Q. Caudron, S. R. Donnelly, S. P. C. Brand, and Y. Timofeeva,
Computational convergence of the path integral for real dendritic morphologies,
The Journal of Mathematical Neuroscience, 2 (11), 2012.

The Motif method presented in Chapter 5 was developed in collaboration with Jamie Harris, of Warwick Complexity, whose insight and experience with Green's functions brought us to the idea of motifs on trees.

This thesis has not been submitted for consideration at any other institution.

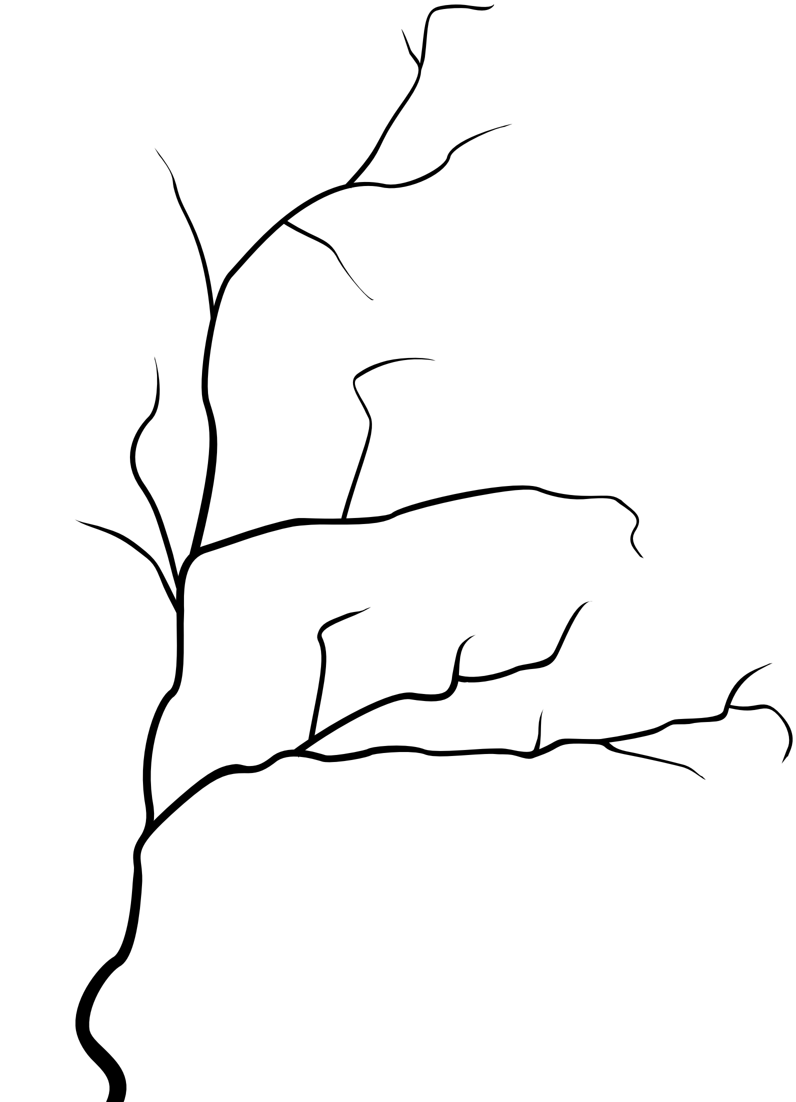
Abstract

When we think about neural cells, we immediately recall the wealth of electrical behaviour which, eventually, brings about consciousness. Hidden deep in the frequencies and timings of action potentials, in subthreshold oscillations, and in the cooperation of tens of billions of neurons, are synchronicities and emergent behaviours that result in high-level, system-wide properties such as thought and cognition. However, neurons are even more remarkable for their elaborate morphologies, unique among biological cells. The principal, and most striking, component of neuronal morphologies is the dendritic tree.

Despite comprising the vast majority of the surface area and volume of a neuron, dendrites are often neglected in many neuron models, due to their sheer complexity. The vast array of dendritic geometries, combined with heterogeneous properties of the cell membrane, continue to challenge scientists in predicting neuronal input-output relationships, even in the case of subthreshold dendritic currents.

In this thesis, we will explore the properties of neuronal dendritic trees, and how they filter and integrate the electrical signals that diffuse along them. After an introduction to neural cell biology and membrane biophysics, we will review Abbott's dendritic path integral in detail, and derive the theoretical convergence of its infinite sum solution. On certain symmetric structures, closed-form solutions will be found; for arbitrary geometries, we will propose algorithms using various heuristics for constructing the solution, and assess their computational convergences on real neuronal morphologies. We will demonstrate how generating terms for the path integral solution in an order that optimises convergence is non-trivial, and how

a computationally-significant number of terms is required for reasonable accuracy. We will, however, derive a highly-efficient and accurate algorithm for application to discretised dendritic trees. Finally, a modular method for constructing a solution in the Laplace domain will be developed.



centre for
complexity
science

Chapter 1

Preface

> “ *It is the brain which is the messenger to the understanding.* ”
>
> - Hippocrates

1.1 The Brain in Context

Neuroscience is the interdisciplinary study of the nervous system, from the molecular biology of the smallest structures in the brain, to the emergent functional properties of billions of neurons, wired intricately together. The immense scales spanned by the brain, both in space and time, present a major challenge for neuroscientists. Experiments can be performed at the sub-cellular level, with the help of ever-developing imaging methods or ultra-sharp electrodes, assessing the dynamics of ions or molecules within the cell; to the macroscale, where neuroscience seeks to understand behaviour, memory and consciousness. As such, neuroscience spans a very large range of spatial scales, as well as a broad spectrum of scientific fields, existing at the interface between biology, chemistry, physics, computer science, mathematics, psychology and medicine. Many interdisciplinary approaches are possible, allowing a number of angles of attack for the modelling of neuronal processes, from mechanistic to phenomenological. Many biophysical models, such as the Human Brain Project [Markram, 2012], attempt to include as much biological realism as is known, constructing neural circuits in a bottom-up approach, from the molecular level. This typically comes at a large computational cost, as the system size is scaled up : even running on some of the world's most powerful computers, the project's predecessor, the Blue Brain Project [Markram, 2006], had simulated a cortical column of ten thousand neurons in 2008, and a mesocircuit of only one million neurons by 2011 – a small fraction of the human brain's nearly 10^{11} neurons [Herculano-Houzel, 2009].

In contrast, models focusing on simplification of the biological processes in favour of mathematical efficiency are able to simulate far larger systems. These approaches abstract away the biological detail, using functional forms to approximate the system's dynamics instead of fully simulating the system. This is exemplified by Izhikevich's 2005 simulation, in which a system of 10^{11} neurons and 10^{15} synapses – a system the size of a full human brain – was simulated on the neuronal level, albeit in far less detail than with biophysical models. The simulation ran over fifty days, and provided one second of real data. Izhikevich and Edelman's [2008] later simulation later reconciled the single neuron, cortical columnar, and large-scale white matter spatial scales, informed by experimental data; due to the added complexity, this system only contained one million neurons.

One element brought into Izhikevich's 2008 simulation which was absent from his earlier, larger system, was branching dendritic morphology. The inclusion of spatially-extended dendritic cables introduces a significant computational difficulty : due to their arbitrarily-branching natures, electrical activity on dendritic trees cannot be solved for explicitly, except in very rare cases. As such, dendrites are typically modelled discretely in space, and the voltage across their membranes computed numerically, which can be very slow.

In the following chapters, we address the question of dendritic computation in the context of passive dendrites. Our aim is to compute the impulse response kernel for the stimulation of a dendritic tree, which provides a backbone for signal integration in the brain. Once known, the impulse response kernel can be used for the rapid evaluation of signals diffusing along complex branching structures, directly in the time domain. We will review various mathematical methods published in the literature, and attempt to address potential improvements, especially in terms of computational efficiency and ease of algorithm implementation. Using analytical techniques and computational algorithms, then, we will develop novel methods applicable to linear cable theory problems, focusing largely on the path integral approach derived by Abbott *et al.* [1991], and on its computational convergence. We will demonstrate that the errors in the calculation can be highly unpredictable, and that it is non-trivial to construct the solution in an efficient manner, whether the terms be ordered randomly or by heuristic. We also study a method for computing the solution to cable theory problems in the frequency domain, providing analytical expressions requiring numerical inversion from the frequency domain. These results all provide a means of computing the impulse response function of a dendritic system, providing insight into the dynamics of current flow along the dendritic tree and allowing the efficient computation of the voltage response anywhere on the tree, as a result of current input.

1.2 Thesis Outline

Chapter 2

We begin by a short history of the field of neuroscience, from our knowledge of it in antiquity, and through its major developments in the twentieth century, when neuronal cable theory was born. We then introduce the general biophysics of the neuron, focusing on typical morphology and the excitable properties of its cell membrane. We discuss aspects of neuronal communication, including the generation of action potentials, how synapses transmit signals from one neuron to another, and some of the possible methods used by neurons to encode information. After a review of the history of dendritic modelling, and of modern literature on methods for approaching dendritic cable problems, we finish with a few brief examples of where dendrites may perform computations.

Chapter 3

After a review of important concepts in integral transforms and linear systems, the linear cable equation is introduced, along with the assumptions made in its derivation, some of its steady-state solutions, and the general time-dependent solution on the infinite cable. We then present a summary of Abbott *et al.*'s [1991] path integral for dendritic trees, a method for computing the transmembrane voltage on arbitrary branching structures; we discuss rules for constructing the infinite series solution and derive a novel proof of its theoretical convergence, making fewer restrictive assumptions than in the proof provided by Abbott [1992].

Chapter 4

Using Abbott *et al.*'s [1991] path integral, we construct novel analytical solutions to the Green's function for some simple, symmetric branching structures. We then present our core algorithms for computing the sum-over-trips series solution on arbitrary trees. We begin with methods that use a length-priority heuristic, deriving a language-theoretic approach to constructing the terms in the series as an improvement on the algorithm provided by Cao and Abbott [1993], and then using a k-shortest paths algorithm for efficient implementation. We compare Cao and Abbott's [1993] Four Classes trip ordering with our Length Priority ordering of terms in assessing the computational convergences of the algorithms. A Monte Carlo method is then developed, ordering trips according to a probabilistic heuristic. Finally, we draw on methods from graph theory to derive an efficient algorithm for grouping trips by discretised lengths in order to construct the series solution using blocked terms.

Chapter 5

The methods and algorithms presented in the previous chapter were approaches to computing the dendritic path integral, a convergent time-domain series solution. Here, we consider the exact solution to cable theory problems in the Laplace domain, by constructing a linear system of equations based on graphical motifs. We derive a forward method which requires the inversion of a motif matrix to obtain the coefficients of the Laplace-domain solution, which is easily computationally-implemented.

Chapter 6

Finally, we summarise our results and discuss the potential for future developments and applications, with respect to the proposed algorithms and methods.

Chapter 2

Of Neurons and Dendrites

> “*Most of the brain consists of “wires” [...] The connections as a whole define the information content of the system.*”
>
> - Campbell [1989]

2.1 A Brief History of Neuroscience

The brain is arguably the most complex object in the known universe. At one time thought to be a continuous mesh of tissue, we now know that the human brain contains on the order of one hundred billion neurons. For every neuron, there are ten neuroglia, cells responsible for modulating neurotransmission, amongst other functions. From this enormous, intricately interconnected network of highly-nonlinear units, somehow, emerges consciousness, memory and emotion.

For thousands of years, scientists have tried to understand how the brain functions. The first reference to the brain, in any written text and language, is found in the Ancient Egyptian medical text, the Edwin Smith Surgical Papyrus, written around 1500 BCE. It describes the effect of brain trauma on the rest of the body, recognising that paralysis, contralateral motor control, and sensory perception can be influenced by brain and spinal cord injury. Later, Hippocrates first described the brain as the seat of intelligence within the body. From that point, contributions to our knowledge of brain anatomy came from Herophilus and Galen, and then during the Renaissance, from Andreas Vesalius, Leonardi da Vinci, René Descartes, and many others. It was with the invention of the microscope, however, and the pivotal work of Camillo Golgi on his *reazione nera*, later named the Golgi stain, that individual neurons were discovered. This brought on the formation of the neural doctrine, proposed by Santiago Ramón y Cajal, that the brain consists of a large number of individual functional units, called neurons. Experiments pioneered by Luigi Galvani developed our knowledge of excitable tissue such as muscle, and in the late 19th Century, neurons were shown to be electrically excitable and that their electrical states were correlated with those of their neighbours. As research methods improved, we learned much of structural and functional neuroanatomy, furthered by contributions from scientists such as Broca, Wernicke, and Brodmann.

In 1864, Julius Bernstein developed the differential rheotome, an instrument for the quantitative measurement of the neuronal action potential. Decades later, he proposed the hypothesis that excitable cells are surrounded by a selectively-permeable membrane, and explained excitation as an increased permeability to potassium ions. It was only fifty years later, however, that the first mathematical description of the action potential was published. Work by Ostwald, Cole, Curtis, Katz, and many others, culminated in a series of papers by Alan Hodgkin and Andrew Huxley on a set of nonlinear ordinary differential equations, describing the initiation and propagation of action potentials; this work earned Hodgkin and Huxley the Nobel Prize in Physiology or Medicine in 1963. Today, the Hodgkin-Huxley model – and related conductance-based neural models - are still extensively used in the modelling and simulation of networks of neurons where biophysical realism is required. Less realistic, the simplest model neuron is arguably Lapicque's integrate-and-fire cell, describing neurons as a simple capacitor circuit. Appreciated for its mathematical and computational simplicity, it has also spawned many variants, such as the leaky integrate-and-fire, as well as nonlinear (quadratic, exponential, adaptive) versions.

All of the aforementioned neuronal models assume that neural cells have no spatial extent. Whilst a large part of modern work in theoretical neuroscience is on the network level, with many interconnected neurons, it remains that neurons are connected by axons and dendrites - long, cable-like branching structures which carry electrical signals from one neuron to another. The manner in which dendrites branch dictates how incoming signals are integrated and sent to the soma, and influences how much impact a presynaptic action potential will have on the postsynaptic neuron's membrane potential. The groundbreaking work of Wilfrid Rall, on the application of cable theory to neural fibres, is the basis for modern theoretical work on signal propagation along branching cables. The voltage along a spatially-extended neuronal structure can be calculated or simulated using one of many approaches derived from Rall's models and methods, with cable theory at their heart.

Passive cable theory, the linear framework adapted to dendrites by Rall, allows mathematically-tractable insight to be made regarding the dendrites' spatiotemporal filtration properties on signal integration. It provides a means to tackle cable problems for large, complex structures, such as those representative of neuronal morphology, enabling us to replicate or explain experimentally-verified findings. Cable theory models the continuum flow of ions along one-dimensional cables, according to a set of assumptions regarding the typical dimensions, chemical composition, and electrical properties, of dendritic fibres.

Cable theory provides us with the ability to compute the spatiotemporal filtering properties of a tree's morphology. This fundamental characteristic can provide a great deal of information regarding how a particular neuron integrates electric currents : the impulse response function, which encodes the system's reaction to an external stimulus, can tell us the amount of delay or attenuation that a signal will incur as it diffuses along the dendritic cables. It can aid us in understanding how a number of signals from different parts of a dendritic tree are integrated and filtered when they reach the soma. Due to this signal filtration, understanding the link between dendritic morphology and the signals observed at the soma as a response to stimulus is essential in the study of neuronal input-output relations.

The manner in which the brain processes information remains a mystery. In the past, simple computations were attributed to individual neurons; it was assumed that a network as complex as the brain could produce significant emergent behaviour, accounting for higher levels of information processing, from perception to emotion. Incoming signals would be summed, or integrated, and if a threshold was met, the neuron would fire a spike. By adding dendrites into this picture, we immediately have available a larger number of computational tools, from nonlinear summation in active dendrites, to resonance, to delay and linear signal filtration in passive dendrites [Spruston *et al.*, 1999]. Today, we understand the importance of dendrites in neuronal computations and the additional dynamics they bring to communication in the brain. There is therefore a great deal of information to be found in a neuron's morphology.

2.2 Morphology of Neurons

Neurons are electrically-excitable nerve cells found in the nervous systems of all Eumetazoa – all animals except sponges and other very simple species. These highly-specialised cells process and transmit information between one another and to the rest of the body, via a complex and elaborate branching network of filamentous neural processes, the axons and dendrites. The human brain, approximately 1200 cm^3 in volume [Cosgrove *et al.*, 2007] and weighing 1.5 kg [Parent and Carpenter, 1995], contains on the order of 10^{11} individual neurons [Herculano-Houzel, 2009], each with an average of 10^4 connections to other neurons [Drachman, 2005], local or distal, forming anything from small, sparse clusters to enormous, densely-connected hypercolumns.

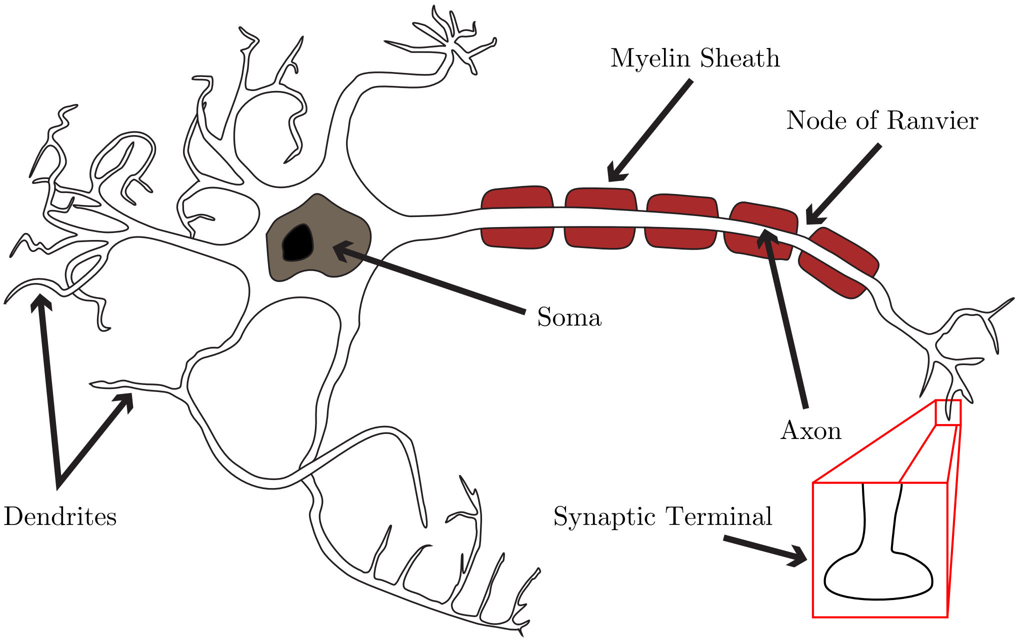


Figure 2.1: A caricatured neuron, showing the key components of neuronal structure and morphology.

Structurally, neurons show a beautiful range of diversity and morphology, and there is really no such thing as a "typical" neuron. In general, however, neurons can be broken down into three parts : a cell body or *soma*, a long, thin axon, and a number of branching dendritic trees. They vary considerably in size and shape between species, area of the brain, and type of neuron. However, what truly typifies neurons are their dendritic trees. To say that dendrites exhibit diverse morphologies would be a gross understatement – dendrites may be highly-selective in their branching, perhaps connecting to only a few specific targets, or may be dense and space-filling, such as the cerebellar Purkinje cells, which sample from tens of thousands of parallel fibre axons. Figure 2.1 shows a schematic illustration of a neuron, demonstrating the three parts that generally make up a neuron : the soma, the axon, and the dendrites.

2.2.1 Soma

Neuronal cell bodies exhibit a great variety of sizes. Some of the largest known somas belong to the sea slug *Aplysia california*, and can approach close to 1 mm in diameter [Gillette, 1991] – although they are usually much more compact. The soma houses a large number of organelles, such as the nucleus and Nissl granules, which are important sites for protein synthesis, particularly the synthesis of neurotransmitters. Other synthesis also largely occurs inside the soma : the sugars and lipids that make up the intracellular fluid, or cytoplasm, are made here.

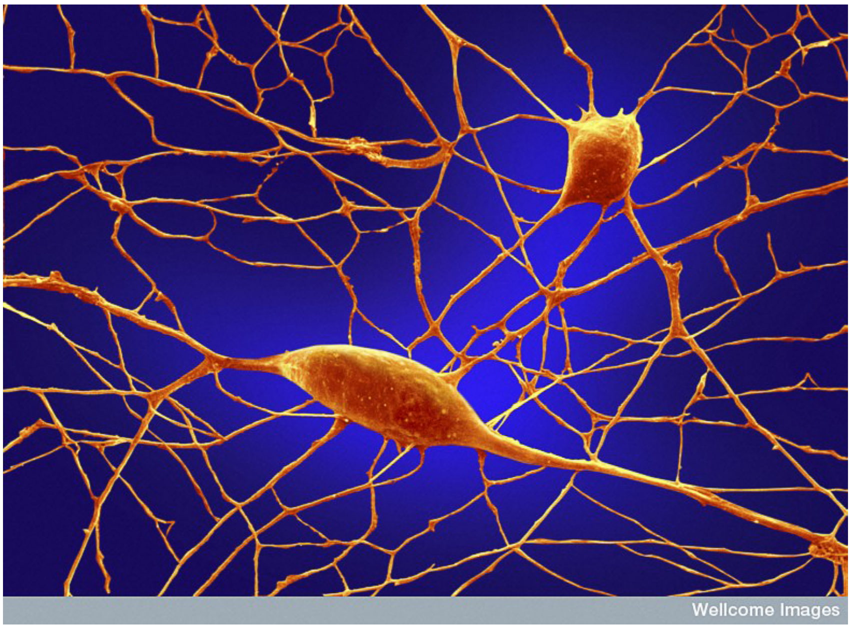


Figure 2.2: Scanning electron micrograph of differentiating Purkinje neurons. Once their development is complete, these neurons will become some of the largest in the brain. *Annie Cavanagh, Wellcome Library, London.*

Like all biological cells, the soma is enclosed in a cell membrane, a selectively-permeable barrier through which flows of ions and small molecules are controlled. The output pole of a neuron, the axon, extrudes from the soma at the axon hillock; a soma can also support a number of dendritic trees, projecting out from the soma in a manner similar to the axon. Figure 2.2 shows a scanning electron microscopy (SEM) image of two developing neurons, with mature somas but incomplete axons and dendritic trees.

2.2.2 Axon

The axon is a long projection that can extend great distances, generally branching extensively as it meets other neighbouring (or distal) neurons. It connects to the cell body at the axon hillock, a specialised part of the soma that swells to become the axon. Its high density of ion channels makes it capable of generating action potentials, which then propagate down the axon, away from the soma, usually to other neurons but also to muscle or gland cells. Axons are typically wrapped in myelin sheaths, composed of an electrically-insulating material which reduces membrane capacitance while increasing its resistance, leading to a smaller loss of current through the membrane. The small spaces between the myelin sheaths, the nodes of Ranvier (see Figure 2.1), reveal the uninsulated axonal membrane, which is capable of high levels of electrical activity. The action potential is continuously regenerated at these points, where electrical signals are amplified by the triggering of further action potentials. Hence, the axon supports the faster saltatory propagation of action potentials.

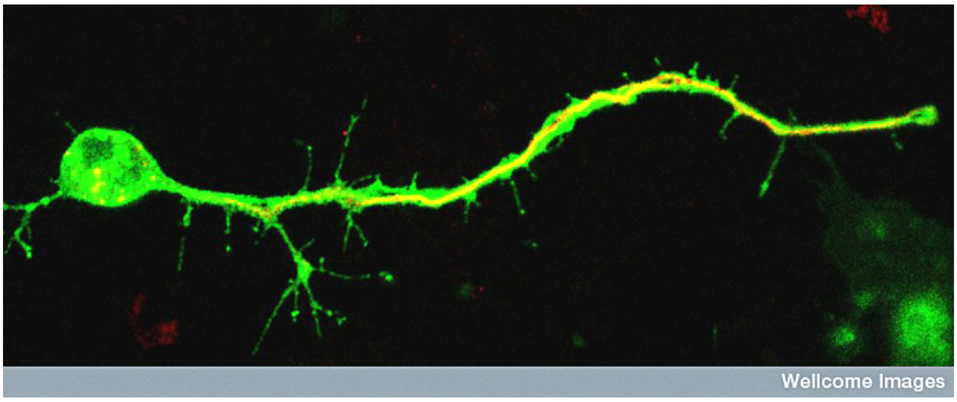


Figure 2.3: Confocal micrograph of a *Drosophila* neuron growing *in vitro*. The axon can be seen emerging from the soma at the axon hillock, retaining a fairly constant diameter as it branches into smaller protrusions. *Guy Tear, Wellcome Library, London.*

The axon is typically around 1 μm in diameter and, unlike dendrites, does not taper. It can reach over one metre long in humans : from the bottom of the spinal cord to the tip of the toes. The giraffe's primary afferent axons can even reach several metres in length. There exist axonless neurons, such as the olfactory bulb's granule cells. Figure 2.3 shows a confocal micrograph of the soma and axon of the neuron of a fruit fly, *Drosophila melanogaster*.

2.2.3 Dendrites

Dendrites are branching filaments that protrude from the soma. They conduct electrochemical signals, delivered by the axons of presynaptic neurons, to the soma, integrating the input from thousands of connecting neurons, both temporally and spatially. Their geometrical and synaptic properties are known to strongly influence the way in which action potentials (and subthreshold signals) are integrated [Vetter *et al.*, 2001; Krichmar *et al.*, 2002; London and Häusser, 2005]. Passive cable theory can be used to describe the changes in membrane potential at the soma as a function of dendritic geometry (connectivity, branch lengths and radii). Unlike axons, the membranes of dendrites are not protected by myelin, and are covered by ion channels and other transmembrane proteins which may contribute to signal modulation. A particular aspect of such modulation is dendritic democracy [Magee and Cook, 2000; Häusser, 2001], whereby a weak input from the distal part of a dendritic tree can be amplified such that distal dendritic connections contribute as significantly as proximal connections to the somatic potential. Another phenomenon seen in dendrites is the support of back-propagation of action potentials initiated at the soma, depolarising the dendritic tree, and modulating synaptic potentiation or depression.

Whilst axons can travel extremely long distances in cellular terms, dendrites tend to branch out in close proximity to the soma. Typical dendrites are rarely longer than 1 mm in length, and generally taper to much smaller radii towards their distal ends. The taper of dendritic branches is clear in Figure 2.4, which shows a micrograph of a group of cells in the cerebellum.

2.2.4 Diversity of Dendrites

The diversity observed in dendritic morphology is staggering. A neuron with no dendrites, assuming a roughly spherical soma, has very limited surface area onto which connections can be made. Dendrites are able to significantly increase neuronal surface area with little cost to volume; 97% of a motor neuron's surface area is dendritic [Ulfhake and Kellerth, 1981]. Intuitively, therefore, one source of variation stems from the fact that dendritic branching patterns can be related to the mode of connectivity between neurons. For example, Spruston *et al.* [1999] describe the density of dendritic branching as varying with the spatial sampling properties of the tree : from sparse trees that sample from other neurons selectively (Figure 2.5D) to those that sample extensively by filling the space (Figure 2.5A). Pyramidal cells (Figure 2.5E) can be placed somewhere in the middle of this scale; their dendrites

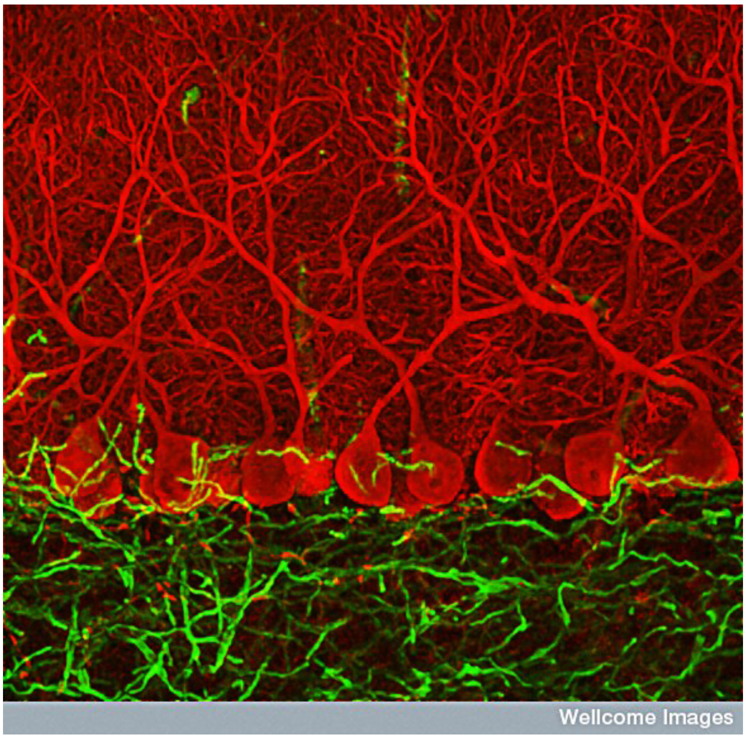


Figure 2.4: Confocal micrograph of cerebellar Purkinje cells (red). Unlike those in Figure 2.2, these cells are fully-developed and display the extensive dendritic branching typical of Purkinje cells. *Ludovic Collin, Wellcome Library, London.*

often sample selectively from different cortical *layers*, at a given distance from the soma, but much less at other distances.

Space-sampling can be directionally uniform, where dendrites radiate in all directions, such as spinal cells or cerebellar granule cells, or can sample preferentially in a specific direction, such as in unipolar or bipolar neurons, in those with conical (mitral cells) or fan-shaped trees (Purkinje cells), in those that sample from a plane (retinal horizontal cells) or even multiple, parallel planes (amacrine cells) [Spruston *et al.*, 1999].

Diversity also strongly varies according to the species. Ramón y Cajal argued that the complexity of dendritic morphologies is correlated with that of the host's nervous system : neurons that need to connect to more neurons of different types, and at an increasing number of different locations, develop more complex geometries.

Moreover, there is a fundamental relation between the structure of a dendritic tree and the function of the neuron. The tree's properties, such as its branching pattern and tapering properties, affect how incoming signals are integrated into the

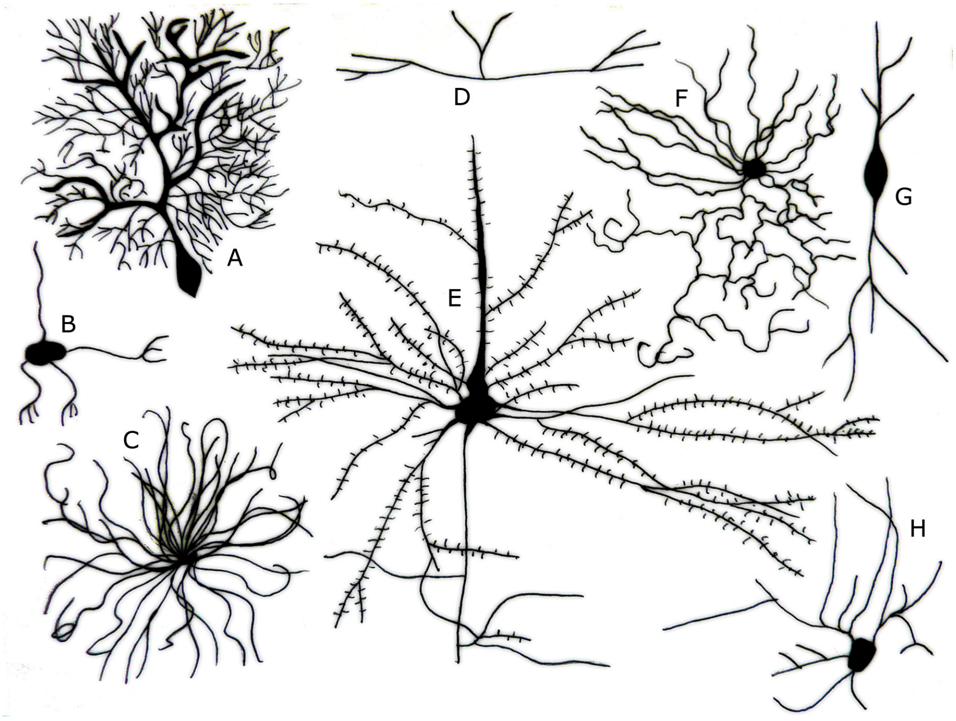


Figure 2.5: A selection of different neurons, demonstrating a wide variety of dendritic morphologies. A : Purkinje cell; B : granule cell; C : motor neuron; D : tripolar neuron; E : pyramidal cell; F : chandelier cell; G : spindle neuron; H : stellate cell. *Illustration by Ferris Jabr, based on the work of Ramón y Cajal. Scientific American, May 2012, reproduced with permission.*

soma. The visualisation of an analogous dendritic tree in *electronic* space is possible via Zador *et al.*'s [1995] morphoelectrotonic transform, a mathematical model describing how the morphology of a dendritic tree influences the attenuation and delay of a presynaptic signal. Using this mapping, it becomes clear that morphology has incredible impact on a tree's signal integration. Work by Vetter *et al.* [2001] showed that dendritic morphology is an important component in determining the efficacy of presynaptic potentials, and how strongly these incoming signals contribute to somatic voltage. The morphology of the dendrites even affects the neuron's own firing pattern [Mainen and Sejnowski, 1996]. The relation between dendritic morphology and computational capability is an important and active field of research in modern neuroscience. With all neurons having a very similar biophysical make-up and excitation mechanisms, it is morphology that truly sets them apart from one another.

2.3 The Biophysics of Excitable Cells

2.3.1 Structure of the Cell Membrane

All cells are enclosed and protected by a cell membrane. The fluid outside of the cell, composed primarily of water with high concentrations of sodium and chloride ions, is kept isolated from the cell's internal environment, a collection of organelles in a potassium-rich solution called the cytosol, by a cell membrane. The cell's membrane is composed of an assembly of phospholipids, molecules which consist of a hydrophilic head and a long, hydrophobic tail. In the presence of polar fluids, such as the extracellular medium or the cytosol, the hydrophobic tails are forced to aggregate together, presenting the charged, hydrophilic heads to the fluid which surrounds them, almost as a shield against the polar water. This self-assembly into a lipid bilayer, by a process of micellisation, is driven by hydrophobic interactions between the fluid and the phospholipids. The result is a protective and isolative membrane, separating the inside of the cell from its external environment. The compositions of the extra- and intracellular fluids endow them with a high electrical conductivity. In contrast, the lipid bilayer, which contains no free ions or charge carriers, acts as an insulator. This enables a potential difference to exist across the cell membrane, such that the outside of the cell is at a different potential to its internal environment. The cell's transmembrane potential is then defined as $V_{\mathrm{m}}(t) = V_{\mathrm{i}}(t) - V_{\mathrm{e}}(t)$, the difference between the intracellular and extracellular potentials. When the cell is at rest, the cytosol is at a lower potential than the extracellular space. A neuron's resting potential is defined as the voltage across the lipid bilayer when the neuron is in dynamic equilibrium, and is typically around $V_{\mathrm{m}} = -70$ mV, although this depends strongly on the particular neuron.

Small molecules, such as oxygen and water, are able to pass through the lipid bilayer by passive diffusion, whereas macromolecules are typically unable to enter or exit the cell due to their larger sizes. The cellular membrane is made selectively permeable to certain macromolecules and ions by a large family of membrane proteins, which can act as passive diffusion channels. These channel proteins are typically formed of multiple subunits that span the membrane in a circular arrangement, leaving a pore through which the relevant species may diffuse more rapidly. Many ion channels have very narrow pores which allow only a certain ionic species to pass through in single file, although nearly as rapidly as they would move in free fluid. The simplest ion channels are passive, open pores, and their concentration along the cellular membrane dictates how readily a specific ion may move down its electrochemical gradient. In the language of electrical components, these passive channels act as fixed linear resistors, and the membrane can be said to have a specific membrane resistance, R_{m}, measured in Ω cm^2. Figure 2.6 is an illustration of a simple ion channel, surrounded by a cell's lipid bilayer to either side.

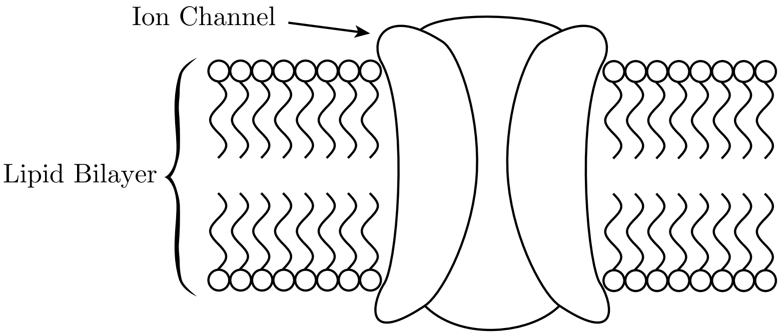


Figure 2.6: A cross-section of a cell's membrane, showing the phospholipid bilayer with an ion channel positioned across it.

Other ion channels are gated, only allowing ions to pass depending on transmembrane voltage, chemical or even mechanical signals. Amongst the passive ion channels, these are essential for correct neuronal function, enabling the rapid voltage changes required to generate action potentials. They may be thought of as state-dependent variable resistors.

While voltage-gated ion channels are able to contribute to the maintenance of an electrochemical gradient across the cell membrane, active ion pumps are crucial for establishing this gradient. These membrane proteins actively move ions across the membrane, *against* the electrochemical gradient, fuelled by energy sources such as ATP. They allow the accumulation of high concentrations of certain ions inside and outside of the cell. Like passive channels, active ion pumps can also be voltage-gated or ligand-gated. By using ion pumps and carefully controlling the permeability of ions across the membrane, a cell is able to maintain a healthy internal environment, using the concentration gradients and potential difference across the membrane to drive reactions that are essential to cellular function. These can be modelled in terms of electrical components by a battery in series with a variable resistor.

With two conductive fluids separated by an insulating membrane, a capacitative element is also introduced to the cell's electrical behaviour. In fact, with the membrane only two molecules thick, the specific membrane capacitance can be relatively high : around $C_m = 1\ \mu$F cm^{-2} at typical transmembrane voltages. The total capacitance, C, is determined by the thickness of the membrane, and in turn determines how much charge, Q, can be built up across the membrane at a given transmembrane potential, according to $Q = CV_{\mathrm{m}}$.

2.3.2 Resting Potential and Equivalent Circuits

At equilibrium, the cell is said to be at its resting membrane potential, E_{m}, with each ionic species contributing a weighted average to the potential, according to the Goldman-Hodgkin-Katz equation :

$$E_{\mathrm{m}} = \sum_x E_x \frac{P_x}{P} \,, \tag{2.1}$$

where E_x is the resting potential of ionic species x (for example, x can be Na^+, Cl^-, Ca^{2+} or K^+), P_x is the permeability of ion x, and $P = \sum_x P_x$. Each ion's resting potential E_x can be found according to the Nernst equation, which relates ionic concentrations on either side of the cell membrane to the ion's electrical charge :

$$E_x = \frac{R\,T}{z\,F} \ln\left(\frac{[x]_{\mathrm{out}}}{[x]_{\mathrm{in}}}\right), \tag{2.2}$$

where R is the ideal gas constant, T is the temperature in Kelvin, z is the integer charge of the ion, F is Faraday's constant and $[x]$ denotes the ionic concentration, either outside or inside the cell. An ion's resting potential determines the magnitude and direction of the ionic current :

$$I_x = g_x(V_{\mathrm{m}} - E_x), \tag{2.3}$$

where g_x is the ion's conductance, and can be fixed by, or a function of, the membrane's permeability to that ion, P_x. At equilibrium, therefore, the cell is kept at its resting potential ($V_{\mathrm{m}} = E_{\mathrm{m}}$) by a balanced flow of different ionic currents, determined by the membrane's permeability to the specific ions and by the driving force, $V_{\mathrm{m}} - E_x$, that they experience due to their own resting potentials. When the concentration of ions either inside or outside the cell changes, whether due to a chemical or electrical signal, the cell experiences changes in the ionic currents that were keeping it in dynamic equilibrium. If, say, the extracellular sodium concentration $[\mathrm{Na}^+]_{\mathrm{out}}$ suddenly increased, then its resting potential, E_{Na^+}, would also increase, leading to a change in the sodium current. Because sodium is in excess outside the cell in normal conditions, the net sodium current flows into the cell from the extracellular medium. In this example, the extracellular sodium concentration has increased, and the magnitude of this current would then also increase, bringing more sodium into the cell per unit time, and the cell would depolarise (become less negative).

With any change to the cell's potential, V_m, a capacitative current is also generated :

$$I_\mathrm{C}(t) = C_\mathrm{m}\,\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}t}. \tag{2.4}$$

The membrane's capacitance determines how quickly a cell's potential can change when a current is presented. Large capacitances result in slowly-changing transmembrane potentials. Together with the membrane's high resistance to transmembrane currents, the membrane's potential can be modelled using a standard resistor-capacitor (RC) circuit. Modellers and electrophysiologists often use electrical circuits as an analogy for a neuron's electrical properties. An equivalent electrical circuit can be made for the model neuron or for a patch of its membrane, allowing its dynamics to be studied without having to model the neuron's individual channels and their properties. Typically, the equivalent circuits consist of a capacitor and a number of resistors in parallel. The capacitor is fixed, being determined by the capacitance of the cell's lipid bilayer, while the resistors can be fixed (for the passive, open ion channels such as the leak current) or variable (for voltage-gated or active channels). A resistor is used for each ionic species that is being modelled, with the magnitude and direction of the current flowing through the resistor depending on the resting potentials and permeabilities of each ion, and can be a function of the neuron's state. Figure 2.7 shows an equivalent circuit for a simple neuron with dedicated sodium, potassium and chloride currents due to active ion pumps, where the transmembrane resistance for each ion is variable.

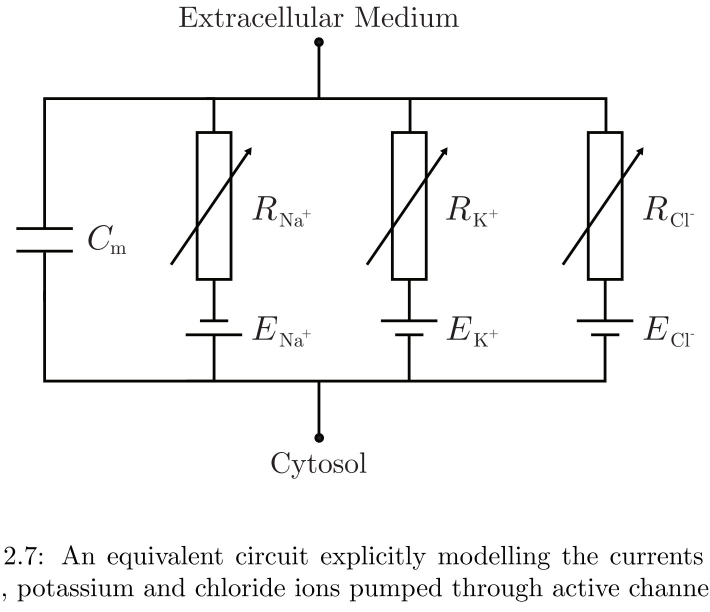


Figure 2.7: An equivalent circuit explicitly modelling the currents due to sodium, potassium and chloride ions pumped through active channels.

Using an equivalent circuit makes it straightforward to model the dynamics of the model neuron. The membrane's resistive current can be described using Ohm's law :

$$I_R(t) = \frac{V(t)}{R}. \tag{2.5}$$

The capacitative component of the RC circuit modelling the membrane can be modelled as in (2.4). Kirchhoff's current law can be used to equate the inward and outward currents, for the resistive component. Applied to the circuit in Figure 2.7, Kirchhoff's current law gives :

$$C_\mathrm{m}\,\frac{\mathrm{d}V_\mathrm{m}(t)}{\mathrm{d}t} + \frac{V_\mathrm{m}(t) - E_{\mathrm{Na}^+}}{R_{\mathrm{Na}^+}} - \frac{V_\mathrm{m}(t) - E_{\mathrm{K}^+}}{R_{\mathrm{K}^+}} - \frac{V_\mathrm{m}(t) - E_{\mathrm{Cl}^-}}{R_{\mathrm{Cl}^-}} + I_\mathrm{inj}(t) = 0, \tag{2.6}$$

where $I_\mathrm{inj}(t)$ is any current injected directly into the neuron by a transmembrane microelectrode. The simplest neuronal model, the leaky integrate-and-fire neuron, also has the simplest equivalent circuit : a capacitor in parallel with a single resistor. If we set $E_\mathrm{m} = 0$ for simplicity, and assume that initially, $V_\mathrm{m}(0) = 0$, then the equation governing its dynamics is

$$C_\mathrm{m}\frac{\mathrm{d}V_\mathrm{m}(t)}{\mathrm{d}t} + \frac{V_\mathrm{m}(t)}{R} + I_\mathrm{inj}(t) = 0, \tag{2.7}$$

and the time-dependent solution is

$$V_\mathrm{m}(t) = R_\mathrm{m}\, I_\mathrm{inj}(t)\left(e^{-t/\tau} - 1\right), \tag{2.8}$$

where $\tau = R_\mathrm{m}C_m$ is the membrane's time constant. The membrane's potential therefore reacts exponentially, with time constant τ, to discontinuous changes in injected current, $I_\mathrm{inj}(t)$. The time constant τ is typically around 10 ms, although, like the resting potential E_m, this is highly dependent on the type of neuron.

RC circuits are linear systems. This important property forms the fundamental basis of the work in this thesis, the core of which is dedicated to computing the *impulse response kernel*, a function that fully characterises the system's response to an injected stimulus. Linearity allows a system to act as a passive filter, altering the frequency content of an injected signal. In actual fact, there is evidence that dendritic voltage may depend nonlinearly on transmembrane currents, and that signals propagating along dendritic trees can be boosted by active processes (for a review of some of the mechanisms involved, see [Johnston *et al.*, 1996]). Whilst active conductances like ion pumps are inherently nonlinear systems and contribute considerably to neuronal input-output relations, it is important to recognise that the

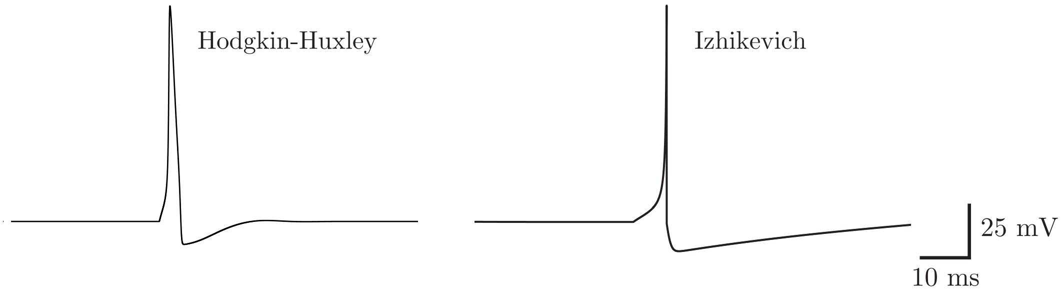


Figure 2.8: The shapes of two action potentials, as simulated by the Hodgkin and Huxley [1952] conductance-based model (left) and the phenomenological Izhikevich [2003] model (right).

passive properties of dendritic membranes provide the fundamental core for signal filtration and integration, and thus remain an essential component in understanding electrical signalling in neural systems, and provide the underlying mechanisms for neuronal communication.

2.4 Neuronal Communication

2.4.1 Action Potentials

An action potential is a rapid rise and fall in electrical membrane potential. Also known as spikes, action potentials are extremely short-lived, typically existing on the sub-millisecond timescale, and travel at speeds varying from one to one hundred metres per second down the axon. They are typically considered "all-or-nothing" events, in that spikes are generated when the transmembrane potential exceeds a particular threshold, and not otherwise, and that the magnitude of the spike is independent of the injected current. Whilst the shape of an action potential can be highly variable depending on cell type, stimulation and environment, the structure can be generalised, as in Figure 2.8.

Due to the high number of inputs to a given neuron, each connected by noisy synapses and with signals arriving continuously from different parts of the dendritic tree, the cell's membrane potential can display strong fluctuations. Despite this, action potentials are extremely strong, sharp signals which stand out clearly from the background when measured close to the soma, as seen in Figure 2.9.

Action potentials are a result of a positive feedback loop between the transmembrane potential and ionic permeability. As a response to an increase in the membrane potential or to chemical stimuli, certain voltage-gated ion channels be-

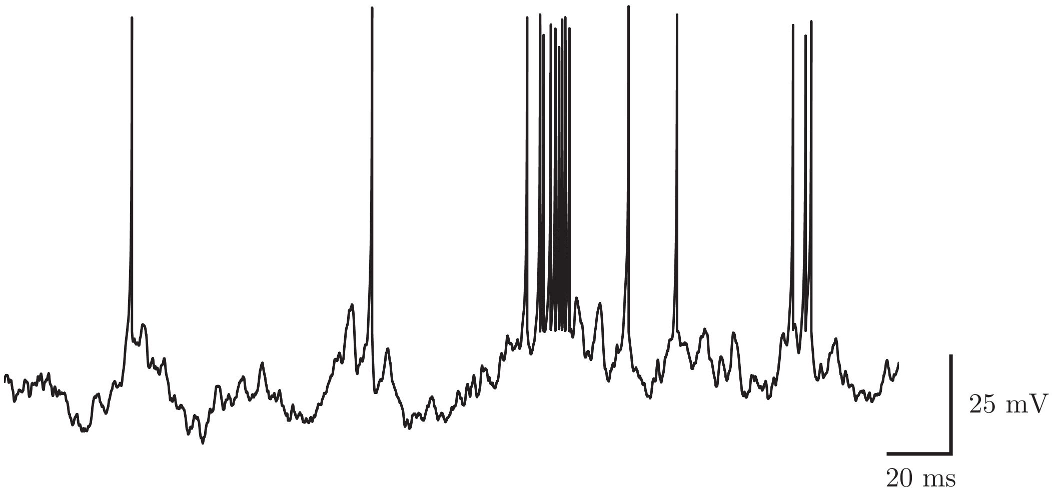


Figure 2.9: A spike train simulated by the Izhikevich [2003] model, driven by an Ornstein-Uhlenbeck noise. For this particular parameterisation, this Izhikevich neuron exhibits a period of bursting as well as phasic spiking.

gin to open, increasing the permeabilities P_{Na^+} and P_{K^+} to sodium and potassium ions respectively, allowing sodium to flow into the cell while potassium escapes it. For small perturbations around E_{m}, the inbound sodium current is overwhelmed by the outbound potassium current, bringing the membrane potential V_{m} back towards its resting value, E_{m}. However, sufficient depolarisation of the membrane potential leads to a further increase in sodium permeability as more voltage-gated sodium channels open. This significantly changes the ion's resting potential, E_{Na^+}, and thus, the membrane's resting potential, E_{m}. This further affects the membrane potential, V_{m}, causing it to rise suddenly.

The positive feedback pushes the membrane potential V_{m} towards E_{Na^+}, at which point all sodium channels are open and sodium permeability is at a maximum. Sodium channels begin to inactivate and close, while further voltage-gated potassium channels open and potassium continues to flow out of the cell, hyperpolarising the cell. In combination with an influx of calcium ions, potassium permeability is exceptionally high and the potential V_{m} is driven past its resting potential E_{m}, towards that of potassium, E_{K^+}, in an undershoot or *afterhyperpolarisation.* The membrane's potassium permeability returns to its normal values as the membrane potential tends to E_{m}. At this point, the cell enters a refractory period during which it cannot fire a spike while its sodium channels are still inactive (as opposed to simply closed) and will not respond to any stimulus. These channels transition back to their closed states after about one millisecond, after which another spike may be initiated. This refractory effect is what enables an action potential to be transmitted in a single direction : the area in front of the action potential contains closed sodium channels, which can be opened normally, whereas the area behind it is in a refractory state due to its sodium channels being inactivated.

In conductance-based models, such as that of Hodgkin and Huxley [1952], action potentials are not generated by a changing resting potential, E_{m}. Individual ionic resting potentials are considered constants; instead, voltage-dependent gating variables are used to describe the extent to which ionic channels are open. Then, an ionic current takes a form modified from that in (2.3). The currents due to sodium and potassium, respectively, are

$$\begin{aligned} I_{Na^+} &= g_{\mathrm{Na}^+}\, m^3\, h\, (V_{\mathrm{m}} - E_{\mathrm{Na}^+}) \\ I_{K^+} &= g_{\mathrm{K}^+}\, n^4 (V_{\mathrm{m}} - E_{\mathrm{K}^+}), \end{aligned} \tag{2.9}$$

where m, $n, h \in (0, 1)$ are dynamic gating variables which evolve as a function of the transmembrane potential V_{m}.

Action potentials are typically initiated at the axon hillock, and then travel down the axon to the presynaptic boutons, carrying a message to the upstream component of the synapse. There, the message can be sent to any postsynaptic neurons by means of a chemical or electrical signal.

2.4.2 Synapses

Synapses are the sites at which neurons exchange signals. Typically directional, synapses are locations where the membranes of two neurons come into close proximity, allowing either the diffusion of chemical signalling molecules, or the direct flow of ions, between the membranes. Each of the human brain's 10^{11} neurons connect to other neurons an average of 10^4 times, although this number varies enormously, depending on the type of neuron. The vast majority of synapses are chemical in nature. These consist of a presynaptic *bouton*, or terminal (see Figure 2.1), directly opposite a region of the postsynaptic cell that is rich in neurotransmitter receptor proteins. When the synaptic terminal of the presynaptic neuron receives an action potential, an influx of calcium is caused through the opening of ion channels in the terminal's membrane. This causes an activation of the proteins attached to neurotransmitter vesicles inside the synaptic terminal, causing them to dock onto the cell membrane, and to release their neurotransmitter contents into the space between the two neurons, the *synaptic cleft.* The neurotransmitter molecules diffuse across the space, and bind to receptors on the postsynaptic terminal's membrane. This causes the membrane permeability to certain ions to change, causing a change in the postsynaptic neuron's local transmembrane potential. Depending on the type of neurotransmitter released, and thus the ion channels opened, the resulting postsynaptic potential change can be either excitatory (depolarising) or inhibitory (hyperpolarising). The reuptake of neurotransmitter back into the presynaptic neuron by active pumps terminates the signalling process.

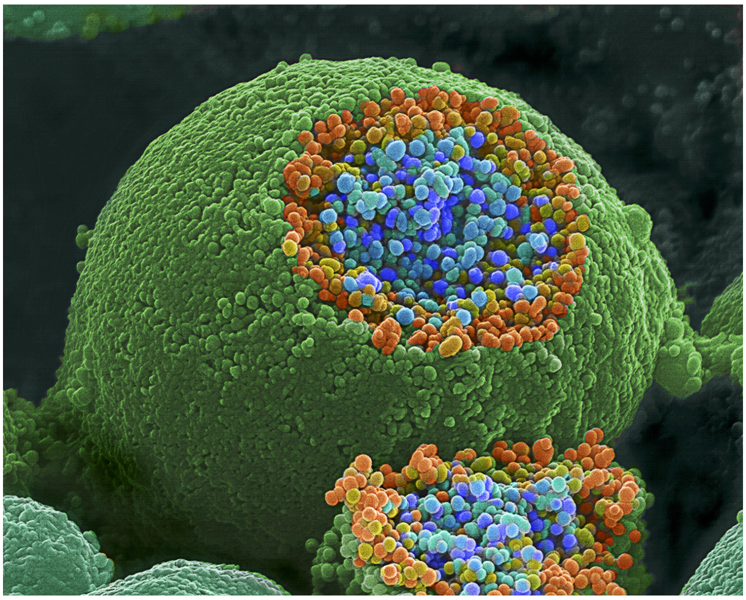


Figure 2.10: Scanning electron micrograph of a synapse, with part of the membrane removed to reveal synaptic vesicles, in orange and blue. *MicroAngela, Biological Electron Microscope Facility, Pacific Biomedical Research Centre, University of Hawaii at Manao.*

Chemical synapse behaviour can be modulated by spike activity. Depending on spike timing, intensity and firing frequency, the quantity of neurotransmitter released may change, postsynaptic membrane receptors may develop augmented or reduced sensitivity, and ion channel concentration may be altered [Ho *et al.*, 2011]. Both positive and negative feedback loops exist, whereby synaptic strength is controlled on both short and long time scales. For example, synapses may be potentiated for up to a few minutes, following a stimulation by repetitive pulses, a phenomenon induced in a form of plasticity called post-tetanic potentiation [Zucker and Regehr, 2002]; long-term potentiation and depression are the primary mechanisms for modulated synaptic strength, occurring over several hours to days in mammals [Glanzman, 2010]. Spike-time-dependent plasticity is a process whereby two highly temporally-correlated spikes from different neurons affect the strength of synapses between them [Abbott and Nelson, 2000]. Together, these forms of synaptic plasticity are thought to be possible underlying mechanisms for memory and learning in the brain [Elgersma and Silva, 1999]. Figure 2.10 shows a chemical synapse and vesicles full of neurotransmitter.

Electrical synapses, also called gap junctions, occur less frequently than chemical synapses, and are predominantly found in the retina and cerebral cortex. Gap junctions consist of a large number of transmembrane proteins which cross the membrane of two cells, with a channel diameter of typically under 2 nm, simply forming a pore between two neurons. This allows both ions and smaller signalling molecules to flow from one cell to the other, bidirectionally. In further contrast to chemical synapses, gap junctions do not have gain, and hence cannot amplify a received signal. However, the much smaller distance between neurons, on the order of 3.5 nm, and the lack of a cascade of events to reconstruct the signal, mean that gap junctions are much faster synapses than chemical synapses. They have a typical delay of around 0.2 ms, an order of magnitude shorter than for chemical synapses. The speed at which signals can move through gap junctions means that they allow many neurons to fire synchronously; they are often found in escape mechanisms such as in *Aplysia*'s danger response system, where a large amount of ink is quickly released. Despite their extreme simplicity, in comparison to chemical synapses, there is evidence for long-term regulation in gap junctions, such as in the modulation of retinal sensitivity during light and dark adaptation [Hu *et al.*, 2010]. Like chemical synapses, the permeability gap junctions can be modulated by voltage [Mammano, 2006], or by neurotransmitters [Cachope *et al.*, 2007]. In theoretical modelling studies, gap junctions have been successfully modelled as time- and state-dependent ohmic resistors [Baigent *et al.*, 1997].

Synapses, then, are essential to neuronal communication : they are the bridge between single neurons and the network level. They provide a level of tuneable control over the extent to which signals are passed between neurons, both linearly and nonlinearly, and are at the heart of network-level plasticity.

2.4.3 Network Connectivity and Structure

The brain is far from a uniform mesh of neurons. It is composed of a large number of heterogeneous structures, arranged hierarchically by substructure, and interconnected both locally and distally by projecting axon bundles. The cerebral cortex, for example, is a sheet-like region of tissue, between two and four millimetres thick, covering the outermost part of the cerebrum. It is made of up to six horizontal layers of pyramidal neurons with unmyelinated axons, with layers differentiated by the type of neurons and the destinations of their axons. Cortical columns are formed vertically by neurons in different layers with near-identical receptive fields, a spatial region where a stimulus is likely to alter the firing of a neuron.

Neuronal network connectivity can be studied on many scales, the smallest of which is the network topology of neurons connected by their axons and dendritic trees in full detail. The structure of dendrites is known to strongly influence neuronal computation [Mainen and Sejnowski, 1996; Vetter *et al.*, 2001]. The passive properties of dendrites lead to a spatiotemporal filtration, where any signal is generally attenuated as a function of distance travelled. It has been speculated that a neuron may therefore label incoming signals as a function of their delay and shape. Neuronal receptive fields are dictated by dendritic morphology, and the signals received are filtered by the properties of the dendritic trees. A sparse tree may well carry a signal more faithfully than a heavily-branching tree such as a Purkinje cell, with less current diffusing along other branches. In addition to their role in integrating synaptic inputs via passive cable-like properties, as well as the active qualities that enable them to regenerate propagating signals, dendritic trees are known to be able to generate back-propagating action potentials, which cause a depolarising current to travel up the dendritic tree [Stuart *et al.*, 1997].

Connectivity can also be considered on higher spatial scales. The neocortex, for example, is arranged in vertical hypercolumns, or groups of approximately 60,000 neurons with nearly identical receptive fields. On yet larger scales, anatomists organise the brain into structures called *nuclei* which, typically, operate together in a functional manner. The brain is divided into large areas such as the frontal, temporal, occipital, and parietal lobes, the cerebellum, the brainstem, and the basal ganglia, each of which can be further divided into areas with functional similarities. The posterior part of the frontal lobe, for instance, houses the motor strip, which produces movement, while the hippocampus is associated with long-term memory formation. Figure 2.11 shows a magnetic resonance imaging (MRI) scan of a central coronal section through an adult human brain, with certain anatomical features of the occipital and temporal lobes labelled.

Despite an average of 10^4 connections per neuron, each differing in strength over time, and even with connections being made and destroyed continuously, the general topology of the brain's connectivity remains fairly static, except in rare cases. In examples of traumatic injury, where a region of the brain is damaged or removed, evidence points towards the reorganisation of connectivity in order to bypass this region and allocate its tasks to other regions.

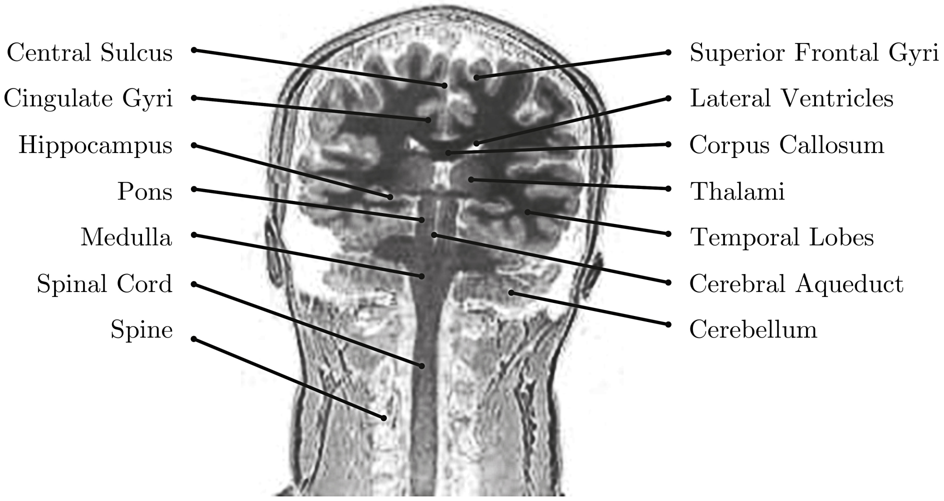


Figure 2.11: Coronal-section structural MRI scan of the author's brain, showing various neuroanatomical areas. *With thanks to Tomohiro Ishizu at the Wellcome Neuroimaging Lab at UCL.*

2.4.4 Plasticity

Neuroplasticity is a term used to refer to the structurally dynamic processes in the brain, where neural pathways are rewired or synapses are strengthed or weakened, in response to stimuli or environmental changes. On the large scale, cortical remapping is the massive rewiring of connectivity in the brain, typically in reaction to an injury, or removal or death of a part of the brain. This is routinely seen in the surgical treatment of severe epilepsy, for example, where seizures are seen to be localised in a specific region of the brain and do not otherwise respond to medication. The surgical removal of the damaged part of the brain can lead to a partial reduction or a complete elimination of future seizures; it can, however, leave the patient with neurological issues such as paralysis, impaired vision, or speech and language issues, in varying levels of severity. Functional recovery of these processes occurs as the brain remaps its connections to avoid the damaged areas and reform functional networks. The restriction of this type of plasticity has been shown to reduce recovery of sensory and motor function [Thallmair *et al.*, 1998].

Plasticity also occurs at synapses, which can be strengthened (postsynaptic potentiation) or weakened (postsynaptic depression), resulting in a change in the amplitude of postsynaptic action potentials. Plasticity on this scale is closely associated with learning and memory. The classic Hebbian theory that "neurons that fire together, wire together" states that when two neurons fire simultaneously, synapses between them are reinforced. It is used as a basis for explaining associative learning, where two cells serving related functional roles may be stimulated at the same time.

Synaptic plasticity is typically broken down into short-term and long-term effects. Short-term synaptic facilitation may arise from an increased number of vesicles present in the presynaptic bouton, or an increased probability of vesicle release, while short-term synaptic depression can be caused by a depletion in the number of neurotransmitter vesicles available due to recent excessive spiking. Described as lasting on the order of several seconds, short-term plasticity is thought to contribute to temporal filtering of incoming signals, because it is elicited by temporal activity patterns [Fortune and Rose, 2000].

Plasticity on timescales of hours or days is referred to as long-term plasticity, and is closely associated with the formation of memories or consolidation of learning. Long-term potentiation and depression can be explained by the idea of spike-time-dependent plasticity. Backed by experimental results, spike-time-dependent plasticity leads to the potentiation of a synapse if the presynaptic neuron fired just before the postsynaptic neuron, and a depression of the synapse if the presynaptic neuron fired just after. In this way, if the presynaptic neuron was likely to contribute to the postsynaptic neuron's excitation, that connection is strengthened, while the influence of signals that did not cause excitation is reduced. This process can be related back to the theory of Hebbian learning : if a neuron is downstream (postsynaptic) of another (presynaptic) neuron, and it fires in response to an incoming action potential fired by the presynaptic neuron, these can be said to have "fired together", and the synapse is potentiated. If the presynaptic neuron fired after the postsynaptic cell, this may be explained by coincidence rather than the passing of a meaningful signal, and the depression of the synapse can lead to an increased signal-to-noise ratio. Compelling evidence for spike-time-dependent plasticity in dendrites is summarised in a recent review by Dan and Poo [2004], where this form of plasticity is associated with learning and memory.

2.5 The History of Dendritic Physiology and Modelling

The modelling of dendritic cables has its roots in nineteenth-century electrophysiology, when experimentalists such as Matteucci and du Bois-Reymond first began measuring the electrical currents generated in muscle tissue in the 1840s. The idea of a core conductor, a long and thin electrically-conducting core surrounded by a thin, insulating membrane, was first put forward by Matteucci [1863]. The idea was to be refined by Hermann in a series of publications over the next thirty years, during which Weber, a colleague of Hermann, developed a mathematical treatment of core conductors [Weber, 1873] which describes the flow of current through a long, three-dimensional structure. Early testing of core conductor theory was qualitative in nature, until the necessary equipment and preparation methods had been developed.

Significant progress in microscopy drove advances in physiology and microanatomy in the 19$^{\text{th}}$ Century. It was Theodor Schwann who put forth the idea of "*one universal principle of development for the elementary parts of organisms*" [Schwann, 1839] and hence, the theory that cells form the basic "units" in all living things, and that new cells are produced from preexisting cells; in spite of this, the view was not readily accepted with respect to the nervous system due to its anatomical complexity. In 1860, the basic anatomical structures of neurons – the soma, the axon, and the dendrites – had been described by the German neuroanatomist Otto Deiters [1860]; however, a mischaracterisation contributed to further evidence for *reticular theory*, the preferred theory that the nervous system's protrusions, the axons and dendrites, fused together seamlessly to form a continuous reticulum. The theory was particularly strongly supported by Camillo Golgi, an Italian physician who had developed a groundbreaking histological staining technique [Golgi, 1873] which allowed the sparse staining of entire neurons using a silver chromate precipitate.

A young Spanish anatomist, Santiago Ramón y Cajal, discovered the staining method in 1887 and, instantly attracted to its practicality, began work on improving it. In 1888, Ramón y Cajal began a systematic histological study of the vertebrate nervous system that culminated in his challenging of reticular theory in 1894 after a series of pioneering, radical publications, which were later organised in his magnum opus, *Textura del sistema nervioso del hombre y de los vertebrados* [Ramón y Cajal, 1899]. Ramón y Cajal's work pointed to a theory of individuality of the cells in the nervous system, where axons terminate freely and that information travels from the dendrites and the soma, down the axon (the *formula of dynamic polarization*). Shortly after German scientist von Waldeyer-Hartz coined the term *neuron* to denote the individual cells in Ramón y Cajal's description, the *neuron doctrine* was created; it states that neurons are individual cells which consist of soma, axon and dendrites, and that they conduct impulses in a directional manner.

The work of Ramón y Cajal was not limited to proposing the neuron doctrine, a central description that, updated, still stands today, despite some exceptions. His work with the Golgi stain led to a collection of highly-detailed illustrations of neural morphology and connectivity. This was the first time that neuroanatomy had

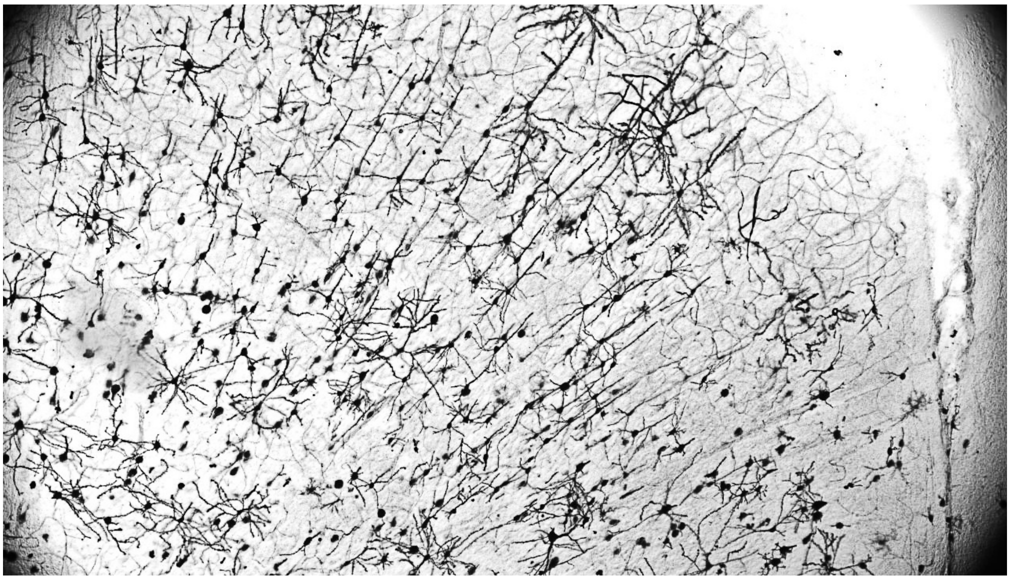


Figure 2.12: Neurons stained by the Golgi technique applied to a 200 μm coronal slice of the rat brain. The stain shows a number of pyramidal cells, with the rest being glial cells. *Image reproduced with thanks to Kyle Ploense, Department of Psychological and Brain Sciences, University of California at Santa Barbara.*

been revealed in such fine detail, demonstrating the enormous variety in neuronal morphology, hinting at the functional differences between different neurons.

Meanwhile, in 1855, a series of letters regarding a submarine, transatlantic telegraph system was presented to the Royal Society. The exchange, initiated by George Gabriel Stokes, pointed William Thomson towards the work of Michael Faraday regarding the bandwidth limitations of the telegraph cable. Thomson promptly derived the cable equation with application to the transatlantic telegraph, describing the dynamics of the cable membrane's voltage as a function of space $-\infty < x < \infty$ and time $t \geq 0$:

$$kc\,\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} - hv, \tag{2.10}$$

where c is the capacitance across the insulation per unit length, k is the resistance of the cable per unit length, and where h parameterises the leak of current through imperfect isolation around the cable. Using the ideas of Fourier [1822], Thomson [1854] provided solutions for both steady-state and transient stimulation of the cable. The work earned him a knighthood the very next year, and in 1892, he was ennobled by Queen Victoria, henceforth to be known as Lord Kelvin.

It was Hoorweg [1898], Cremer [1899], and Hermann [1905] who first recognised the applicability of Thomson's cable theory to neuronal core conductors. The approximation and accompanying simplification of reducing the problem to a one-dimensional cable proved to be both incredibly important and technically sound [Pickard, 1971]. Estimates for the membrane capacitance and resistivity were improved upon using new measurement techniques by Curtis and Cole [1938] and by Cole and Hodgkin [1939] respectively, both in the giant axon of the squid – a very large axon discovered by John Zachary Young [1936]. The axon's large diameter (up to 1 mm) made it possible for experimentalists to insert voltage-clamp electrodes into the axonal lumen. In their 1946 paper, Hodgkin and Rushton fully derived the cable equation and estimated the passive cable parameters from a single axon [Hodgkin and Rushton, 1946]. The next year, Davis and Lorente de Nó [1947] presented their results on the peroneal nerve of the bullfrog, a nerve bundle comprised of both myelinated and unmyelinated axons, but were unable to accurately estimate the parameters of the cable equation due to their recordings being across several axons with varying radii and levels of myelination. Further papers from Hodgkin [1947] and Katz [1948] made cable theory for single, unmyelinated axons a concrete reality. Several important technical achievements were made in the following years : Marmont [1949] and Cole [1949] developed space-clamping and current-clamping; Hodgkin and Katz [1949] succeeded in isolating the sodium current from the potassium current; and Hodgkin *et al.* [1952] were later able to combine voltage-clamping with space-clamping. These techniques enabled Hodgkin and Huxley [1952] to perform their seminal work on characterising the dynamics of the membrane voltage as a function of these ionic currents – a conductance-based model still in widespread use today, which explained the initiation and propagation of action potentials in neuronal cables. This work earned its authors the Nobel Prize in Physiology or Medicine in 1963.

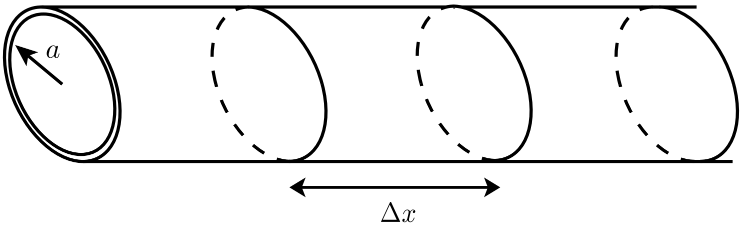
Core conductor theory and cable theory laid the foundations for the modelling of currents in branching dendritic trees. The work of Hodgkin and Huxley brought about an increase in electrophysiological recordings, and advances in technology led to an interest in the axon's unmyelinated cousins – the dendritic trees. In 1957, Wilfrid Rall wrote a letter to Science in which he corrected the underestimates in the membrane time constants made by experimentalists during intracellular recordings with microelectrodes [Coombs *et al.*, 1955; Araki and Otani, 1955; Frank and Fuortes, 1956; Fatt, 1957]. Rall's argument, that current injected at the soma is not confined to the soma, but will escape via the membrane and into the dendrites, explained why the time constants measured in motoneurons was an underestimate; he went on to state that taking the cable-like properties of the dendrites into account was essential to correctly estimating the membrane's time constant. Two years later, Rall published *Branching dendritic trees and motoneuron membrane resistivity* [Rall, 1959], the paper which famously introduced cable theory for dendrites, in a more familiar form, which we present briefly here, to return to in detail in Section 3.1 :

$$\frac{\partial V(x,t)}{\partial t} = \frac{\lambda^2}{\tau}\frac{\partial^2 V(x,t)}{\partial x^2} - \frac{V(x,t)}{\tau}, \quad t \geq 0, \tag{2.11}$$

where λ is the system's characteristic length-scale, and τ is the membrane's time constant. The theory formulated the relationships between the geometry of the neuron and its electrical properties, such as membrane resistivity and capacitance. Rall solved for the steady-state membrane potential for cylindrical branches of arbitrary length and radius, and in 1960, for transient potentials due to injected currents [Rall, 1960]. In 1962, Rall introduces the concept of equivalent cylinders [Rall, 1962a], a class of dendritic branching pattern that permits a direct mapping from the complete dendritic tree to a single, uniform cylinder; with it, he successfully predicted the time-course of the transmembrane potential in a model motoneuron of the cat spinal cord.

Another landmark paper by Rall was published in 1964 : the compartmental model [Rall, 1964]. By segmenting the dendrites into small, isopotential compartments, each with passive membrane dynamics, it is possible to solve cable theory problems with a system of ordinary differential equations. Figure 2.13 shows a segment of nerve cylinder, discretised into isopotential compartments, modelled by the equivalent circuit below it. This paradigm is still the most widely-used today, being the core of popular numerical solvers such as NEURON [Carnevale and Hines, 2006] or GENESIS [Bower and Beeman, 1998]. Based on further work by Rall in the latter half of the 1960s, Jack and Redman derived analytical solutions to the cable equation on finite cylinders for a range of injected stimuli and boundary conditions, using which they suggested a method for estimating the membrane's time constant and the cable's electrotonic length [Jack and Redman, 1971].

Despite significant advances over the past fifty years, there still remain many open questions regarding dendritic function. Due to their sheer complexity, the development of generally-applicable methods for understanding the dynamics of voltage on arbitrary dendritic trees is difficult, and visualisation or interpretation of the results of applying such methods is nontrivial. How the morphology of a dendritic tree affects current propagation and, hence, the transmembrane potential which governs communication and computation in the brain, remains a central question in modern neuroscience.



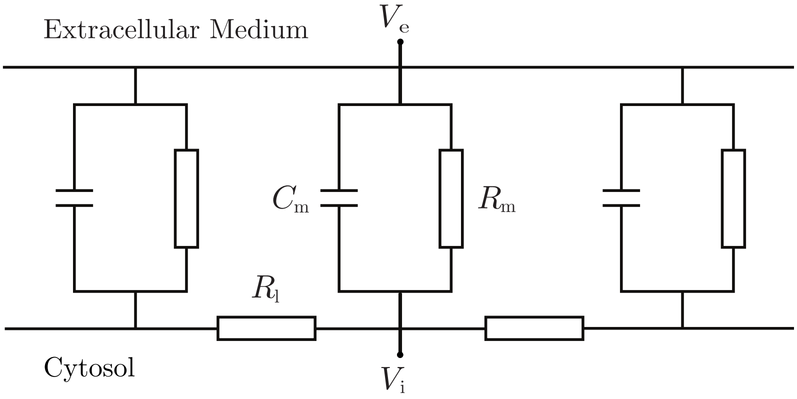


Figure 2.13: A segment of dendritic fibre, modelled as a uniform, insulated cable of radius a and compartmentalised into small, isopotential segments of size Δx. Underneath, the equivalent circuit which encodes for the passive dendritic cable. The main components are the membrane capacitance C_{m}, the transmembrane resistance R_{m} and the longitudinal resistance through the cytosol, R_{l}. Here, three compartments are shown.

2.6 Dendritic Computation

Arguably the biggest strength and weakness of passive cable theory is, simultaneously, the fact that it assumes that the transmembrane potential is a linear function of current. This simplifying assumption has allowed significant progress in the study the voltage on dendritic trees, and is correct should the voltage stay far from the neuron's firing threshold; action potentials are highly-nonlinear events, and they would not be treated correctly by a linear theory. In addition to this, a growing body of evidence is demonstrating that dendrites are endowed with *active*, or nonlinear, properties [Johnston and Narayanan, 2008], attributed to the voltage-dependent ion channels that pepper their surfaces. Despite this limitation, the passive component of current diffusion along dendrites remains the underlying backbone for how current propagates and is integrated along the dendrites.

The core of this work will explore methods in passive cable theory, which aims to understand the fundamental core behaviours of diffusing current on branching structures. Nonetheless, it is important to stress that dendrites are not purely current-integrating devices, but that they demonstrate incredible biological properties, all essential to the correct functioning of the brain, and all allowing the dendrites to apply some sort of transformations to the signal – a mechanism for *computation*. In this section, we briefly describe some of these properties, with the aim of demonstrating areas of future work in the field of dendritic dynamics, and, of course, to expose the potential for computation in dendritic trees.

2.6.1 Spatiotemporal Filtering

An assumption inherent to the modelling of dendrites as core conductors, also present in dendritic cable theory, is that dendritic fibres behave like poorly-insulated electrical cables. Any current in a cable will diffuse along it, leading to a "smoothing" of the signal : if we inject a delta pulse of current, after some time, we will observe a Gaussian (whose width increases with the root of the time spent diffusing). Hence, one property of passive dendritic filtering is the spatial smoothing of the signal. This also implies that a temporal delay occurs between the time of current injection and the point at which the signal is at its maximum, for any distance greater than zero. The leaky property of the membrane bestows an attenuative property to the signal's filtration, as a function of the distance it travels. Passive dendritic transfer functions show a monotonic decay from zero frequency as in a lowpass filter, demonstrating that the attenuation is also a function of the frequency of the signal [London and Häusser, 2005].

As a result of these linear spatiotemporal filtering properties, a signal measured at any distance from its point of injection will be significantly broader than the original current injection, and its integral will be smaller due to some current leaking into the extracellular space. Wilfrid Rall [1964] postulated that this mechanism could be used to perform simple computations, via a "labelling" of the synaptic inputs as a function of their electrotonic distance from the soma.

With the help of the impulse response function, certain characteristics of the spatiotemporal filtration that occur in dendritic trees can be brought to light. Cook *et al.* [2007] were able to measure the characteristic responses of a dendritic tree experimentally by injecting long-duration white noise, and then modelled the tree's response as a linear filter followed by nonlinear static-gain. Their linear filter, equivalent to our Green's function, shows bandpass behaviour induced by the active properties of the dendritic tree; they compare this with the lowpass behaviour seen in passive dendrites. Mainen and Sejnowski [1996] showed, through simulation on reconstructed neocortical neurons, that varying dendritic geometry is sufficient to reproduce the breadth of firing patterns seen in the cortex. Vetter *et al.* [2001] demonstrated how the invasion of action potentials from the soma into the dendritic tree depended strongly on the morphology of the tree, experimentally obtaining visual representations of the mapping from the tree's geometry to its electrotonic properties, similar to the attenograms provided by the morphoelectrotonic transform due to Zador *et al.* [1995].

2.6.2 Spines

Dendrites are not simply long, smooth cables. Instead, most are covered in dendritic *spines*, protrusions dispersed along the dendrites, many of which have thin necks and bulbous heads, much like a presynaptic terminal (see Figure 2.1). Potentially numbering thousands per dendritic tree, dendritic spines serve to receive incoming connections from the axon terminals of presynaptic neurons. Figure 2.14 shows an abundance of spines on a dendritic cable, and demonstrates the high concentration of glutamate receptors on their top surfaces, kept adjacent to the presynaptic terminal by the shape of the spine [Nimchinsky *et al.*, 2002].

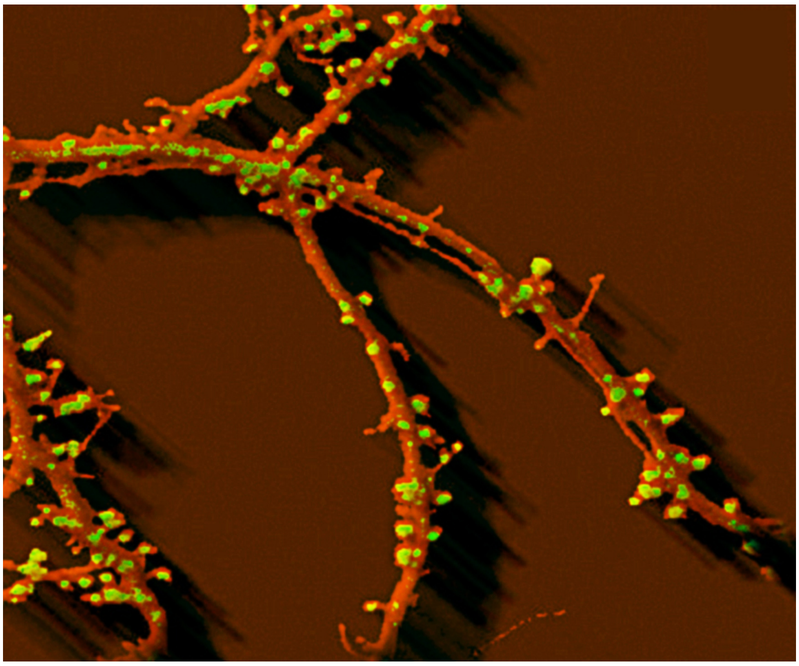


Figure 2.14: A hippocampal neuron expressing the red DsRed protein, with its GluR1 glutamate receptors tagged with the green fluorescent protein, GFP. *Eduard Korkotian, Weizmann Institute of Science, Israel, with thanks.*

The space inside the dendritic spines is full of organelles that play an active role in shaping synaptic plasticity, as well as playing roles in communicating action potentials to the dendrites. For example, there is evidence to show that diffusional exchange between spine and dendrite is one hundred times slower than expected for free diffusion [Sabatini *et al.*, 2002], pointing towards the spine neck providing a means of isolation between the spine and the dendrite on a timescale closer to that of biochemical reactions than purely electrical ones. The neck could also act on faster timescales as a simple Ohmic resistance, which could then be regulated by the length and breadth of the spine neck [Crick, 1982]. Increasingly, spines are being thought of as the dynamic, structural backbones which allow both biochemical and electrical modulation of excitatory synapse strength [Lee *et al.*, 2012].

2.6.3 Active Currents

Should a presynaptic terminal receive an action potential, causing a release of neurotransmitters across the synaptic cleft, a postsynaptic action potential will be initiated in a postsynaptic dendritic spine. With the density of spines able to approach ten per micrometre of dendritic cable on some neurons [Koch, 1999], it is easy to imagine that the current due to this action potential could diffuse along the dendritic cable and push neighbouring spines over their own voltage thresholds, initiating further action potentials [Shepherd *et al.*, 1985]. Because the dendritic action potentials that make up the active currents in dendrites are strongly nonlinear, they cannot be treated by passive cable theory. Baer and Rinzel [1991] studied a system of dendritic cables with a continuous density of spines, assuming that spine head voltage is a function varying continuously with space as well as time. However, in reality, these active currents are generated at discrete, separated sites. A discrete "Spike-Diffuse-Spike" model was therefore proposed by Coombes and Bressloff [2000, 2003], and extended by Timofeeva *et al.* [2006], work in which they observe that the propagating wave is saltatory rather than smooth.

A simpler treatments of active currents was proposed by Koch [1984], who linearised the currents due to the voltage-gated ion channels in dendritic membranes. He modelled the membrane in terms of resistances and capacitances, as in passive cable theory, but also in terms of inductances, a contribution from the linearised active currents. The behaviour of these *quasiactive* dendritic cables demonstrated voltage overshoots in response to step currents; their transfer functions contained a maximum at non-zero frequencies, exhibiting a bandpass quality which selectively amplifies signals with a given frequency. Koch was able to numerically invert the transfer function to obtain the Green's function, which, in contrast with purely passive cables, is no longer positive everywhere.

The additional behaviour brought on by the active channels present on dendritic spines, namely the active propagation and regeneration of action potentials as they diffuse along the tree, and the oscillations around the overshoots, influence the potential of dendrites to perform computational operations. The tree can now be thought of as partially constructed of an excitable medium, with the potential for propagation failure [Timofeeva *et al.*, 2006] due to channel refractory times; certain inputs may now be amplified more than others. Dendritic action potentials are free to travel both up and down the tree, with this backpropagation acting as a positive feedback loop : if a presynaptic potential is enough to send the neuron over its threshold and it fires a spike, this can be sent back up into the tree and impact, via plasticity, the strength of the synapse that received the presynaptic potential [Magee and Johnston, 1997].

2.6.4 Coincidence Detection

Coincidence detection is a phenomenon particularly important to the auditory and visual systems. Intra-aural time differences can be used to locate the source of a sound. Localisation is done by the bipolar coincidence-detector neurons in the brainstem, each of which receives input from both ears into separate dendritic trees, and is able to compare the delay between inputs up with an accuracy of 10 - 100 μs [Agmon-Snir *et al.*, 1998]. When two signals arrive at the soma from different dendrites, an action potential is fired. However, two signals along the same dendrites are integrated in a sublinear fashion, and cause only a subthreshold response at the soma. These neurons therefore demonstrate a maximal response when they receive inputs from both ears.

An analogy can be drawn between coincidence detection and a multiplication operation, given binary inputs, or a logical AND operation. This phenomenon was shown experimentally by Polsky *et al.* [2004], where evidence of a positive feedback loop was demonstrated in pyramidal neurons, for synchronous inputs on the same dendritic branch. The resulting current from NMDA receptor activation would recruit further NMDA channels, leading to a nonlinear feedback, and hence, to coincidence detection.

Backpropagation of action potentials plays a role in modulation of synaptic plasticity, and is mediated by the coincidence detection that occurs at the synapses. Stuart and Häusser [2001] showed that distal synapses on the apical dendrites of some pyramidal neurons could trigger a nonlinear amplification of backpropagating action potentials when paired with incoming synaptic input from a presynaptic neuron.

2.6.5 Directional Selectivity

Neurons need not be bipolar in order to compute the direction in which a signal is coming from. In the retina, starburst amacrine cells, such as those in Figure 2.15, are essential to the selective response of the neighbouring retinal ganglion cells. While the exact mechanism remains uncertain, and despite their symmetry, experimental evidence points to starburst amacrine cells serving a key element in the computation of direction in moving stimuli [Yoshida *et al.*, 2001].

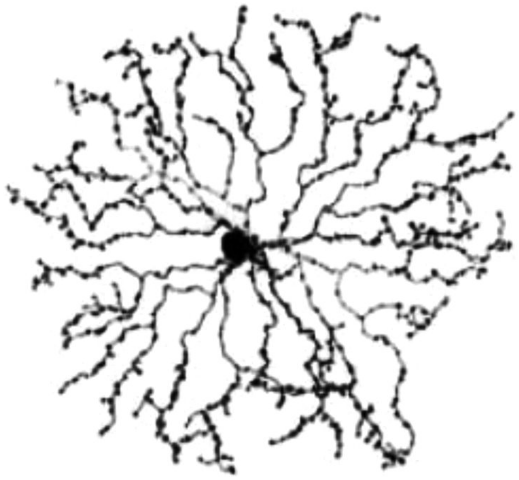


Figure 2.15: A retinal starburst amacrine cell from a mouse, imaged by confocal microscopy. *Reproduced from Keeley* et al. *[2005], with thanks to Patrick Keeley.*

These peculiar cells are one of a small number of neurons that secrete both an excitatory and an inhibitory neurotrasmitter. They respond more strongly to signals moving from the soma towards the distal dendrites, rather than signals moving in the opposite direction [Vaney *et al.*, 2012]. It has recently been shown that starburst amacrine neurons generate directionally-selective inhibition into the ganglion cells, with a postulated mechanism having to do with the large area of overlap between starburst cells and the selective expression of either excitatory or inhibitory neurotransmitters, based on the direction in which the asymmetry of connection between the starburst cell and postsynaptic ganglion cells [Taylor and Smith, 2012].

2.6.6 Dendritic Democracy

Due to the significant spatiotemporal filtration that occurs as current diffuses along dendritic cables, there can be substantial attenuation of a signal from a distal synapse. In pyramidal cells, simulations have demonstrated a hundred-fold attenuation [Stuart and Spruston, 1998], which leads to the distal dendrites having a far smaller impact on somatic voltage than proximal dendrites. Häusser [2001] proposes three possible mechanisms by which "dendritic democracy" could be achieved : amplification of distal signals by increased voltage-gated ion channel density or by scaling the strength of distal synapses, or an increased number of synapses to presynaptic neurons at the distal dendrites. Timofeeva *et al.* [2008] explore a democratic dendritic system analytically, and find that the scale of a synapse must scale linearly with distance from the soma at the proximal dendrites, but superlinearly at the distal dendrites, but that after a threshold distance, there is no scaling of synaptic strength that can ensure democracy for synaptic input.

2.6.7 Computing with Dendrites

In addition to the aforementioned phenomena, which bestow computational power upon the brain, there are a plethora of others, from logical operations to signal segregation, many of which are reviewed in an excellent publication by London and Häusser [2005]. All of these mechanisms contribute to the functioning of the brain, to memory, to consciousness. For many, we are still in the very early stages of study, with little knowledge of how they operate; many others may well still be unknown to us. In order to develop an understanding of these processes, it is essential that we have efficient computational and mathematical tools at our disposal – tools that enable us to directly study the impact of changes in dendritic structure on their current-integrating properties. In the next section, we review a number of elegant and powerful approaches that have been developed to date.

2.7 Continuous-Space Dendritic Modelling

By far the most prevalent methods for the explicit spatial modelling of dendritic trees used today are derivatives of Rall's [1964] compartmental model. The assumptions made by certain geometrical simplifications, such as Rall's [1962a] equivalent cylinder simply do not satisfy the majority of dendritic morphologies; with the exponential increase in computing power over the last decades, the major computational costs associated with the detailed morphological description of a reconstructed dendritic tree are becoming less significant. It is now possible to model highly-complex dendritic trees, with thousands of branches, using numerical simulators such as the aforementioned NEURON and GENESIS packages in short times and to high accuracy, if the number of compartments is large enough. This factor is arguably the greatest bottleneck in the numerical simulation of large neuronal systems, where a substantial number of cells is simulated in full spatial extent. Such simulations, where numerical integration must be performed for an incredible number of differential equations, remain prohibitively slow. Recent work into the development of parallelised, high-performance methods to overcome this issue and allow the simulation of blocks of neural tissue have yielded positive results, with, for example, the successful simulation of a system of one million neurons, discretised into one billion compartments [Kozloski and Wagner, 2011]. Despite these strong advances in efficiency, the authors estimate that it would take approximately one full week of computing for three seconds of simulated data, on a hypothetical massively-parallel computer with 10^7 nodes, for a system the size of the human brain. They suggest that near-realtime simulation of a system this large would require an exaflop computing system.

In contrast to numerical methods, analytical solutions are exact, and obtaining quantitative results from these equations is immensely more efficient, in terms of computational performance. In addition, the possibility of obtaining an analytical solution which can provide mathematical insight into the relationships between model variables and parameters is highly attractive. The difficulty, however, lies in the construction of the analytical solution. Dendritic systems are characterised by asymmetry, inhomogeneous branches and a large number of boundary conditions. Analytical solutions have been found in the time domain only for passive dendrites; resonant dendrites can be approached only in the frequency domain. An essential tool exploited in the construction of analytical solutions for the cable equation on dendritic systems is the Laplace transform, used as a method of solving partial differential equations. The power afforded by this integral transform is such that the vast majority of continuous-space modelling methods for cable theory are Laplace-domain approaches. A number of these methods, able to construct analytical solutions to the cable equation on trees with arbitrary morphologies, will be described below.

2.7.1 Analytical Approaches to Cable Problems

Considerable work has gone into developing approaches that allow the algorithmic construction of an exact, computationally-efficient solution, given a certain dendritic morphology. The vast majority of those published operate from within the Laplace domain : regardless of the difficulties that arise from inverting this transform, the potential for algebraic manipulations that it offers are extremely seductive. Here, we will describe a number of methods for constructing analytical solutions to cable problems on arbitrarily-branching trees, only one of which operates directly in the time domain. The various approaches can be broadly categorised according to their basic mechanism : those based on an influential geometric notation, those which relax the constraints in Rall's [1962a] equivalent cylinder, those which evaluate an infinite sum whose terms must be determined, and those which simply construct a large system of linear equations.

Graphical Calculus

The first general analytical solution to the cable equation on arbitrarily-branching trees was provided by Butz and Cowan [1974]. Citing the difficulty in deriving analytical expressions for voltage transients on branching cables, and the enormous complexity of such expressions when they could be found, Butz and Cowan [1974] derived a graphical calculus based on a geometric notation. Their method allowed them to construct solutions to the cable equation in a piecewise manner, by following a set of rules applied to the tree's morphology. Their method combined the frequency-domain solutions for the response to stimulus on a finite, non-branching cable, and for the voltage around a single branching point. By combining these equations and exploiting symmetries in subgraphs in order to break more complex structures down into either single cables or branching points, the authors developed a method which allows the Laplace-domain transmembrane potential to be written down immediately, for any configuration of stimuli and for any branching tree, including those with a higher branching degree than seen in dendrites. They provide examples of the application of the graphical calculus for branching trees with up to seven branches, which demonstrate how rapidly the manual construction of a solution can become complicated, even for small trees. They go on to consider various boundary conditions and the inclusion of a soma, modelled as a lumped impedance. However, the solutions provided by this method remain in the Laplace domain, a notoriously difficult transform to invert numerically. Butz and Cowan [1974] did not provide details on how to approach inverting their solution into the time domain; however, they show that, for steady-state solutions (taking the limit as $s \to 0$), an inverse transform is not required and the algebraic solution is provided immediately from the Laplace-domain solution.

This difficulty was addressed by Horwitz [1981], who noted that all solutions provided by the Butz and Cowan [1974] calculus can be written as the product of four functions : one each for the measurement point and the injection point, one corresponding to the tree's morphology, and one for the Laplace-transform of the injected current. Horwitz [1981] notes that, should the tree contain certain symmetries, the four functions can be inverted into the time domain analytically, and by convolution theorem, an integral involving the inversions of the four Laplace-domain functions can be used to compute the time-domain transmembrane potential. Horwitz [1981] calls these inverted functions for symmetrical trees "primitive integrals", for which we assume that the tree's branches have the same radii and that the membrane time constant remains constant everywhere along the tree. These primitive integrals are derived for simple structures, and provide a basis for computing the time-domain solution. For asymmetrical dendritic trees, such as those with unequal radii, Horwitz [1981] computes correction terms to each of the four functions using a Taylor expansion, allowing an approximate solution to be found. The time-domain solutions provided by this method are valuable; however, the method remains a manual one in that the inverse transforms need to be found by inspection, from tables relating an expression to its Laplace transform. Even for the primitive integrals, which do not require an approximation by Taylor expansion, finding the inversions in closed-form to allow a convolution to be taken analytically means that this would be non-trivial to automate algorithmically. At least one of the four functions scales in its complexity with the size of the tree, and therefore, there is no guarantee that an algorithm could be written to decompose the Laplace-domain solution from the Butz and Cowan [1974] calculus into Horwitz's [1981] four simpler functions, that a symbolic engine could provide an inverse Laplace transform, or that it could compute a closed-form convolution with the inverted functions in reasonable time.

The work of Butz and Cowan [1974] was taken in another direction by Koch and Poggio [1985], who developed four rules with which the geometric calculus could be applied to any acyclic branching structure algorithmically. The rules describe the frequency-domain solutions for the impedance of a branch and a terminal, the total impedance of a branching point, the voltage along a cable if a current were to be injected at a terminal, and the voltage along a cable should the voltage be known at one end of the cable. This reformulation of the original geometric notation, applied recursively, results in the "folding" of a branching tree into a finite cable, upon which the transfer function can be calculated. Koch and Poggio [1985] went on to implement their rules in an algorithm, thus allowing a solution to be obtained more efficiently than an implementation of the original Butz and Cowan [1974] method. They suggest that the transfer function can be inverted back into the time domain using an inverse Fast Fourier Transform; errors are thus only introduced in this final stage. Despite this, they provide no numerical results demonstrating the method's accuracy. Furthermore, whilst using an inverse Fourier transform can be used to perform an inverse Laplace transform after a variable change, this method can present some inherent instabilities and high sensitivity to parameters, especially in functions with rapidly-changing derivatives, such as those that would arise in the transfer function between two closely-positioned points.

Linear Systems of Laplace-Domain Equations

Citing the long, complicated approaches of Butz and Cowan [1974], Horwitz [1981], and Koch and Poggio [1985], a simpler approach was proposed by Holmes [1986]. By deriving a general equation for the Laplace-domain transmembrane potential at branching points and terminals on the tree, Holmes [1986] puts forth a method revolving around a single equation describing the voltage at a branching point as a function of the voltage at neighbouring branching points. The scheme is easily applicable by computer to complex dendritic trees : by applying the equation as a boundary condition at each node, a system of simple equations for the Laplace-domain transmembrane potential can be constructed. The method yields $n+1$ equations for a tree with n branches – a large number of equations, but by exploiting the sparsity of the system, the equations can be solved using Gaussian elimination prior to taking a numerical inverse Laplace transform to return to the time domain. This method has the advantage of solving for the transmembrane potential everywhere on the tree simultaneously. By way of comparison, the methods based on the Butz and Cowan [1974] calculus will compute the transfer function for a pairwise combination of input and measurement locations on the tree; should we be interested in how current diffuses along the entirety of the tree, we would require multiple applications of the Butz and Cowan [1974] scheme, one for each input-output pair of interest. However, the immediate downside to the Holmes [1986] approach is that the resulting large system of equations requires evaluating at many frequency values in order to invert the transform.

Infinite Series Approaches

In 1991, the first time-domain solution to the cable equation for arbitrary trees was published in a paper by Abbott *et al.* [1991]. The path integral for dendritic trees was formulated by generalising Brownian motion from a line to the tree, and comes in the form of an integral of this generalised Brownian measure over all possible paths from the point of measurement to the point of injection on the tree. Abbott, Farhi and Gutmann use the method of images to write the Green's function solution for a single branching node in terms of the solution for an infinite cable. They obtain a coefficient, giving a probabilistic weighting to the paths as they pass through or reflect off the node, again based on Brownian transition probabilities. These calculations provide a set of rules for constructing Brownian paths which count towards the measure on the tree. The Brownian paths are then generalised to deterministic *trips* from the point of measurement, x, to the point of current injection, y, going through a given series of nodes on the tree, and in a given order. Each trip has associated with it a coefficient, A_{trip}, which is derived from the probabilistic weightings of the Brownian paths; the length of the trip can be calculated as the sum of the lengths of the edges travelled by the trip. For every trip generated, knowing the trip's length and coefficient is sufficient to compute the Green's function solution as a sum-over-trips. The paper finishes by introducing a reciprocity condition where, if we know the Green's function from x to y, we are able to compute the Green's function from y to x using the radii of the edges on which x and y reside. A follow-up paper by Abbott [1992] formalises the construction of the Green's function as a sum over discrete trips rather than as an integral over probabilistic paths, and presents a series of diagrammatic rules for finding valid trips and computing their coefficients A_{trip} from the tree's morphology. Abbott provides a proof that the sum-over-trip series is convergent, assuming an infinite binary tree where the radii of all branches are equal. Some results for a fifteen-branched tree are presented, showing convergence of the sum-over-trips solution after inclusion of approximately ten thousand terms. Finally, in the third paper in the series, Cao and Abbott [1993] provide an algorithm for the generation of valid trips on a tree, by introducing the concept of trip *classes*, a classification of the trips according to the direction they leave their starting point x and the direction from which they arrive at their termination point y. By using the four classes ideology, it is possible to construct all $x \to y$ trips by inserting what Cao and Abbott term "excursions", which are essentially a deviation from the shortest, most direct trip by visiting either a neighbouring node, or by the repeat visiting of a node already in the trip. By adding more and more excursions, Cao and Abbott generate terms in the sum-over-trips solution in order of increasing length, a heuristic which they justify by demonstrating that trips contribute superexponentially less as their lengths increase. The method is especially accurate for short times, and allows a solution for the transmembrane potential to be found directly in the time domain.

In a series of five papers by Major, Evans and Jack starting in 1993 [Major *et al.*, 1993], an analytical solution for the cable equation on arbitrary trees is presented as an infinite series of exponentially-decaying terms, each with different amplitudes and time constants. The parameters of the equation can be obtained using a recursive algorithm that solves a transcendental equation, explicitly revealing some of the system's fundamental physical constants. In direct contrast with Abbott's approach, this infinite series converges particularly well for late times. Their method generalises to dealing with somatic shunts, simulating the damage caused by the insertion of an electrode into a soma, as well as for voltage-clamped systems. Solutions to the amplitudes are found in the Laplace domain, and inverted using Cauchy residue theorem. The time constants obtained, however, are very sensitive to model parameters, and the method is mathematically very convoluted.

Equivalent Cables

A very different approach was proposed by Poznanski [1991], who extended the equivalent cylinder methodology by Rall [1962a, 1977] to allow for an analytical solution to be found for a tapering cylinder. Rall's original constraint stated that the sum of the 3/2-power of the radii of daughter branches must be equal to the 3/2-power of the radius of the mother branch. For a class of motorneurons, this assumption is relatively well-justified; for other types of neurons, this geometric constraint is not satisfied. Poznanski was able to generalise the equivalent cylinder model to deal with a much larger class of neurons, such as those that satisfy trigonometric, quadratic or exponential relations in their radii. His method maps the tree onto an equivalent tapering cylinder, on which the cable equation can be solved explicitly.

In 1993, Whitehead and Rosenberg [1993] distinguished the equivalent cylinder from the equivalent cable : whilst both can be nonuniform, the equivalent cylinder produces the same depolarisation at the soma as does the original dendritic tree, whereas the equivalent cable allows the transmembrane potential to be computed everywhere on the original tree, and not just at the soma. Proposing a method for the construction of equivalent cables, Whitehead and Rosenberg stress that their approach has more value as a tool for visualising the electrotonic properties of the tree, rather than as a method for solving the Green's function : they provide illustrative results of reducing trees to equivalent cables, and explain how certain aspects of the tree's electrotonic behaviour can be inferred back from its morphology. The work was furthered by Lindsay *et al.* [2001], who suggested a series of transformations which map any uniform branching point on the dendritic tree to an unbranched structure, allowing the construction of equivalent cables from dendritic morphologies. An algorithm was presented in a subsequent publication by Lindsay [2003].

Further Work in Cable Theory Methods

Over the last fifty years, numerous powerful approaches have been developed for computing the membrane potential everywhere along a dendritic tree. Some methods are constrained to certain geometries, limiting their applicability in general cases. Many others are applicable to arbitrary morphologies, but may be mathematically or computationally involved, leading to difficulty in implementing computational methods or in slow algorithm runtime. A large fraction of methods are restricted to the Laplace domain, and thus have to surpass the additional hurdle of requiring an inverse numerical Laplace transform. The most popular methods to date are based on compartmental methods, which operate directly in the time domain, but can be numerically demanding for large systems.

In order to understand the voltage dynamics of large dendritic trees, or even of networks of spatially-extended neurons, efficient methods for computing the transmembrane potential are required. Methods with a set of diagrammatic rules, such as those of Koch and Poggio [1985] for Butz and Cowan's [1974] method, or Abbott's [1992] rules for the dendritic path integral method [Abbott *et al.*, 1991], provide simple instructions which may be readily implemented as computer algorithms.

The latter methodology has the advantage of being rooted in the time domain. Cao and Abbott [1993] provide an algorithm for implementing the dendritic path integral, by sampling paths from the dendritic tree in a generally-increasing order of path length. The path integral method therefore offers a promising starting point in the development of novel, more efficient algorithms for computing transmembrane voltage on dendritic trees, which may allow us to study more effectively how dendritic structure and function are related.

2.8 Conclusions

The fascinating array of computational tasks performed by dendritic trees hints at the importance of branching and connectivity in neuronal networks, and how much these computational operations could contribute to higher function in the brain. London and Häusser [2005] suggest that the ultimate aim in the study of dendrites is to objectively demonstrate that they convey a significant computational advantage to neural systems. Such advances could potentially be applied in artificial neural networks and other aspects of bio-inspired computing. Developing an understanding of how dendritic structure is related to their function also opens possibilities in molecular biology and medicine for the treatment of neurodegenerative diseases and other disorders associated with dendritic structure.

A significant challenge in the study of dendrites has been the technological limitation of accurate recording and imaging of dendritic trees. Novel methods, however, show significant promise : examples are those that make use of high-resolution time-lapse imaging to probe developmental changes in dendritic morphology [Satoh *et al.*, 2012], adaptations on scanning and tunnelling electron microscopies to image the substructures of presynaptic terminals [Horstmann *et al.*, 2012], individual synaptic vesicles imaged by tagging their receptors with fluorescent proteins [Padamsey and Jeans, 2012], and calcium signals simultaneously triggered and imaged in dendrites using holographic light patterning [Anselmi *et al.*, 2011].

With recent improvements in computing power, and with highly-advanced experimental paradigms being developed to support theory, a readily-applicable computational framework for simulating dendritic systems is fundamental to furthering our understanding of dendrites in the context of biological and mathematical neuroscience. Decades of research have brought us closer to elucidating many of the computational mechanisms that are experimentally observed in dendrites, such as coincidence detection [Agmon-Snir *et al.*, 1998] or image smoothing by convolution [Cuntz *et al.*, 2003]. Despite this, the structure-function relationship in dendrites remains a topic which requires much further investigation. Therefore, the modelling of electrical current propagating in dendritic trees remains an active field. A large body of research, centred around the work of Wilfrid Rall, is at the heart of contemporary dendritic modelling, and one of the frameworks based on Rall's dendritic cable theory is Abbott *et al.*'s [1991] dendritic path integral. In the next chapter, we will introduce this framework in the context of passive cables and use it to construct solutions to the cable equation on various dendritic geometries. The end goal is for us to develop novel algorithms and improvements on known methods, in order to construct more efficient computational frameworks for simulating current flow in dendritic trees.

Chapter 3

Linear Cable Theory and the Dendritic Path Integral

Dendritic cable theory is a mathematical framework for modelling the passive flow of electric current through neuronal fibres. Derived from core conductor theory, and therefore making assumptions about the physiological properties of dendritic fibres, cable theory can be derived from the theory of electric circuits, such as Ohm's law and Kirchhoff's law.

By careful application of boundary conditions, cable theory can be used to solve flow problems on elaborately-branching structures, such as those common in dendritic trees. A generalisation of cable theory to such arbitrary branching trees, while simple in itself, tends to yield systems which are very difficult to solve. Even for small, simple trees, calculating an analytical solution in the time domain rapidly becomes extremely involved, with a large number of equations and variables to keep track of. On realistic dendritic trees, the sheer number of branches and boundary conditions mean that alternative methods for finding solutions are required. As noted in Section 2.7, many approaches transform the system into the Laplace domain, where the solution to the cable equation is more manageable. The solution is then constructed using a variety of methods, from reduction of the graph to an equivalent cable [Rall, 1962a, 1977; Poznanski, 1991; Whitehead and Rosenberg, 1993], to using repeat patterns or motifs on the tree to apply set rules for building up an analytical expression [Butz and Cowan, 1974; Koch and Poggio, 1985]. A numerical inverse Laplace transform is then typically required to obtain the solution in the time domain. These inverse transforms are infamous for being numerically challenging. Aside from their considerable computational expense, known inversion algorithms work well for certain functions and not for others, making a generic inversion method impossible. Their numerical stability is, at times, only guaranteed in narrow parameter ranges, and require multiple-precision arithmetic methods for computing terms to adequate accuracy, further slowing the computation.

Under the assumption that the characteristic time constant is the same everywhere on the tree (that is, $\tau_i = \tau$ for all branches i), the Laplace-domain Green's function solution can be inverted analytically back into the time domain. Abbott *et al.* [1991] use the time-domain Green's function in a path integral framework, constructing the solution as an infinite sum of functions of the possible trips between two points on an arbitrary tree.

In this chapter, we derive and describe the cable equation, and consider the assumptions made in its derivation from a biological standpoint. After reviewing some important concepts in linear dynamical systems theory, we provide solutions to the cable equation for steady-state and general cases for single cables. We then introduce Abbott *et al.*'s [1991] dendritic path integral, describe how the terms in the solution are constructed and assess the convergence of the infinite series. We finish by deriving certain closed-form solutions that make use of the path integral for simple branching structures : we derive a simple solution for single finite cables, and then introduce a novel analytical solution for symmetrical star graphs that uses a combinatorial counting scheme to enumerate all possible trips on the graph extremely efficiently.

3.1 The Linear Cable Equation

The cable equation is a linear, second-order partial differential equation. It describes the dynamics of the transmembrane voltage, V_{m}, along an infinite cable assumed to be one-dimensional along the spatial coordinate x. It is given here in a form similar to (2.11), with an added term for an applied current :

$$\frac{\partial V_{\mathrm{m}}}{\partial t} = D\,\frac{\partial^2 V_{\mathrm{m}}}{\partial x^2} - \frac{V_{\mathrm{m}}}{\tau} + I_{\mathrm{app}}, \qquad -\infty < x < \infty, \qquad t \geq 0, \tag{3.1}$$

where $D = \lambda^2/\tau$ is the diffusion coefficient of the current, in $\mathrm{mm}^2\ \mathrm{ms}^{-1}$. In order to derive a fully-parameterised, dimensional form of this equation from the biologically-relevant parameters, we begin by introducing the fundamental quantities associated with cable systems.

3.1.1 A Note on Units

Many derivations in the literature prefer the use of parameters defined per unit length of dendritic cable. For the purpose of this derivation, we will use the fundamental units which relate to the physical properties of the cytosolic or membrane material instead.

Specific Longitudinal Resistivity

The longitudinal resistivity R_{l} is a fundamental property of the cytosol, and is measured in units of Ohm-centimetres (Ω cm). This quantity measures how resistive a piece of material of length L (in cm) and uniform cross-sectional area A (in cm^2) is to the flow of electric current :

$$R_{\mathrm{l}} = \frac{A}{L} R, \tag{3.2}$$

where R is the resistance measured across the component. R, in Ohms, is easily measurable for a piece of cable, and once normalised by the dimensions of the sample, we obtain a property of the cytosolic material, as opposed to a quantity like resistance; a short, wide piece of dendritic cable has a much smaller resistance R than a long, thin piece, but both may have equal resistivity, R_{l}.

The cytosolic resistance per unit length of cable, r_{l}, can be found from the cytosol's specific resistivity by

$$r_{\mathrm{l}} = \frac{R_{\mathrm{l}}}{\pi a^2}, \tag{3.3}$$

where a is defined as the radius of the cable, in centimetres. Then, the resistance per unit length r_{l} has units of Ω cm^{-1}, as expected. This relation states that a piece of cable of fixed length with a larger cross-sectional area has a smaller resistance than one of equal length but smaller area, as expected.

Specific Transmembrane Resistivity

Similarly, our derivation will use a property of the lipid bilayer material, the specific transmembrane resistivity R_{m}, rather than the transmembrane resistance per unit length of cable, r_{m}. The quantity R_{m} is measured in Ω cm^2, and is defined as the resistance to current flow across a section of membrane of a given area :

$$R_{\mathrm{m}} = AR, \tag{3.4}$$

where R is now the resistance measured, in Ω, across a piece of membrane of area A. Intuitively, if we take a larger piece of membrane, we expect the total resistance R to decrease; this can be seen in this equation.

The specific resistivity can be related to the transmembrane resistance per unit length of cable, r_{m}. If the radius of the cable, a, is known, then with its circumference being $2\pi a$, we can calculate the resistance per unit length of cable as

$$r_{\mathrm{m}} = \frac{R_{\mathrm{m}}}{2\pi a}, \tag{3.5}$$

a quantity measured in Ω cm. From this, we see that a cable with a larger radius (thus having more surface area) will have decreased total transmembrane resistance.

Specific Transmembrance Capacitance

The capacitative effect that arises because of the membrane's isolative properties can be measured experimentally. If we normalise the total capacitance measured, C (in Farads), by the area of the membrane, A, then we can calculate a property of the membrane material, the specific transmembrane capacitance C_{m} :

$$C_{\mathrm{m}} = \frac{C}{A}, \tag{3.6}$$

measured in F cm^{-2}. As we expect, a larger piece of membrane is able to distribute charge more easily, and hence, has a larger total capacitance.

This quantity can be related to the capacitance per unit length of cable :

$$c_{\mathrm{m}} = 2\pi a\, C_{\mathrm{m}}, \tag{3.7}$$

where a is, as before, the radius of the cable. The capacitance per unit length of cable is therefore measured in F cm^{-1}. Again, we see that, should we take a piece of cable with a larger radius, its greater surface area would allow more capacitance along it.

3.1.2 Derivation of the Cable Equation

We will derive the cable equation from first principles, outlining any assumptions we make along the way, and with care to define parameters as the physical quantities with biologically-realistic units described in Section 3.1.1, such that our derived result may have immediate biological significance and be applicable to real neuronal systems.

We begin by considering a length of uniform cable that satisfies the description of a core conductor, such as that in Figure 2.13. The cable has a radius a cm, and its external environment is isolated from the cytosol by an imperfect membrane of negligible thickness. We define the potential difference across the cable's membrane as

$$V_{\mathrm{m}}(x,t) = V_{\mathrm{i}}(x,t) - V_{\mathrm{e}}(x,t), \tag{3.8}$$

where V_{i} is the potential inside the cable, and V_{e} the potential of the extracellular medium outside the cell, both in millivolts (mV). Later, as per the convention in the field, we will set $V_{\mathrm{e}}(x,t) = 0$, and hence, V_{m} will describe the transmembrane potential relative to a constant external voltage – one of the assumptions treated in Section 3.1.4. As discussed in Section 2.3, the transmembrane potential has a negative value of around $V_{\mathrm{m}} = -70$ mV at rest and, as in Section 2.4.1, a decrease in the transmembrane potential is referred to as a *hyperpolarisation*, whereas should the potential become less negative, or even positive, then V_{m} is said to be *depolarised.*

Referring to Figure 2.13, the voltage at any point along the cable can be written as a function of the voltage some small distance Δx away, should we assume all resistances to be ohmic. Then,

$$V_{\mathrm{i}}(x + \Delta x, t) = V_{\mathrm{i}}(x,t) - \frac{R_{\mathrm{l}}\,\Delta x}{\pi a^2}\, I_{\mathrm{i}}(x,t), \tag{3.9}$$

where R_{l} is the specific resistivity of the cytoplasm and, as in Section 3.1.1, the total resistance for a piece of cable of length Δx can be expressed as $R_{\mathrm{l}}\Delta x/\pi a^2$.

After rearranging and taking the limit as Δx goes to zero, we get

$$\begin{aligned} \lim_{\Delta x \to 0} \frac{V_{\mathrm{i}}(x+\Delta x,t) - V_{\mathrm{i}}(x,t)}{\Delta x} &= \frac{\partial V_{\mathrm{i}}}{\partial x} \\ &= -\frac{R_{\mathrm{l}}}{\pi a^2}\, I_{\mathrm{i}}(x,t). \end{aligned} \tag{3.10}$$

We must now define the internal current, $I_{\mathrm{i}}(x,t)$. Currents flowing into a piece of cable are dependent on those flowing longitudinally through the neighbouring segment, as well as any currents flowing through the membrane, either naturally (due to a potential difference) or artificially (via an electrode). Hence, by conservation of current, we define I_{i} as the sum of these currents :

$$I_{\mathrm{i}}(x+\Delta x,t) = I_{\mathrm{i}}(x,t) - I_{\mathrm{m}}(x+\Delta x,t) + I_{\mathrm{app}}(x+\Delta x,t). \tag{3.11}$$

Here, the transmembrane current I_{m} is negative because of our convention in (3.8) : a negative transmembrane potential (and hence a negative transmembrane current) arises when a current is moving from outside the cell into the cytosol, as is relevant in the case of this definition. A positive transmembrane current would imply that current is leaving the cell.

The transmembrane current is the sum of resistive and capacitative transmembrane currents. Because all resistances are assumed ohmic, then the resistive current is a rearrangement of Ohm's Law, $V = IR$. The capacitative current is a function of the change in voltage over time. Hence, we define the transmembrane current leaving a piece of cable of length Δx as

$$I_{\mathrm{m}}(x,t) = \frac{2\pi a\,\Delta x}{R_{\mathrm{m}}}\,V_{\mathrm{m}}(x,t) + 2\pi a\,C_{\mathrm{m}}\,\Delta x\,\frac{\partial V_{\mathrm{m}}}{\partial t}, \tag{3.12}$$

where $R_{\mathrm{m}} \,/\, 2\pi a\,\Delta x$ is the resistance across a piece of cable of length Δx, and $2\pi a\,C_{\mathrm{m}}\,\Delta x$ is the capacitance across the membrane of a piece of cable of the same length. We can substitute (3.12) into (3.11) and rearrange, to obtain

$$\frac{I_{\mathrm{i}}(x+\Delta x,t) - I_{\mathrm{i}}(x,t)}{\Delta x} = -\frac{2\pi a}{R_{\mathrm{m}}}\,V_{\mathrm{m}}(x+\Delta x,t) - 2\pi a\,C_{\mathrm{m}}\,\frac{\partial V_{\mathrm{m}}}{\partial t} + I_{\mathrm{app}}(x+\Delta x,t). \tag{3.13}$$

Taking the limit as Δx goes to zero, we get the expression

$$\frac{\partial I_{\mathrm{i}}}{\partial x} = -2\pi a\left(\frac{V_{\mathrm{m}}}{R_{\mathrm{m}}} + C_{\mathrm{m}}\frac{\partial V_{\mathrm{m}}}{\partial t}\right) + I_{\mathrm{app}}(x,t). \tag{3.14}$$

Taking the derivative of (3.10) :

$$\frac{\partial^2 V_{\mathrm{m}}}{\partial x^2} = -\frac{R_{\mathrm{l}}}{\pi a^2}\,\frac{\partial I_{\mathrm{i}}}{\partial x}, \tag{3.15}$$

and substituting (3.14) into this result gives us

$$\begin{aligned}\frac{\partial^2 V_{\mathrm{m}}}{\partial x^2} &= -\frac{R_{\mathrm{l}}}{\pi a^2}\left[\,-2\pi a\left(\frac{V_{\mathrm{m}}}{R_{\mathrm{m}}} + C_{\mathrm{m}}\frac{\partial V_{\mathrm{m}}}{\partial t}\right) + I_{\mathrm{app}}(x,t)\,\right]\\ &= \frac{2\,R_{\mathrm{l}}}{a\,R_{\mathrm{m}}}\,V_{\mathrm{m}} + \frac{2\,R_{\mathrm{l}}\,C_{\mathrm{m}}}{a}\,\frac{\partial V_{\mathrm{m}}}{\partial t} - \frac{R_{\mathrm{l}}}{\pi a^2}\,I_{\mathrm{app}}(x,t).\end{aligned} \tag{3.16}$$

Rearranging :

$$\begin{aligned}\frac{2\,R_{\mathrm{l}}\,C_{\mathrm{m}}}{a}\,\frac{\partial V_{\mathrm{m}}}{\partial t} &= \frac{\partial^2 V_{\mathrm{m}}}{\partial x^2} - \frac{2\,R_{\mathrm{l}}}{a\,R_{\mathrm{m}}}\,V_{\mathrm{m}} + \frac{R_{\mathrm{l}}}{\pi a^2}\,I_{\mathrm{app}}(x,t)\\ \frac{\partial V_{\mathrm{m}}}{\partial t} &= \frac{a}{2\,R_{\mathrm{l}}\,C_{\mathrm{m}}}\,\frac{\partial^2 V_{\mathrm{m}}}{\partial x^2} - \frac{1}{R_{\mathrm{m}}\,C_{\mathrm{m}}}\,V_{\mathrm{m}} + \frac{1}{2\pi a\,C_{\mathrm{m}}}\,I_{\mathrm{app}}(x,t).\end{aligned} \tag{3.17}$$

Equation (3.17) represents the dimensional form of the cable equation. From this, we can extract the lengthscale and timescale with which the cable responds to perturbations, which will allow us to write the equation in a simpler form, as in (3.1).

3.1.3 Characteristic Scales

Lengthscale λ

To determine the characteristic lengthscale associated with the dynamics of the transmembrane voltage, V_{m}, we can consider what happens to the voltage when the system is at a steady state on an infinite cable. By applying a constant current $I_{\mathrm{app}}(x) = \sigma\,\delta(x)$ at $x = 0$ for all time $t \geq 0$, and after a sufficiently long period of time (during which time we see transients), the cable will reach an equilibrium distribution of current in space. We can then describe the transmembrane potential, $V_{\mathrm{m}}(x,t)$ as a function of simply space, $V_{\mathrm{m}}(x)$. At this stage,

$$\frac{\partial V_{\mathrm{m}}}{\partial t} = 0, \tag{3.18}$$

for all space. The voltage will then reach a steady state $\lim\limits_{t\to\infty} V_{\mathrm{m}}(x,t) = V_{\mathrm{m}}(x)$, which will satisfy the ordinary differential equation,

$$\frac{a}{2\,R_{\mathrm{l}}\,C_{\mathrm{m}}}\,\frac{\mathrm{d}^2 V_{\mathrm{m}}(x)}{\mathrm{d}x^2} - \frac{1}{R_{\mathrm{m}}\,C_{\mathrm{m}}}\,V_{\mathrm{m}}(x) + \frac{1}{2\pi a\,C_{\mathrm{m}}}\,I_{\mathrm{app}}(x) = 0, \tag{3.19}$$

The general solution to the homogeneous equation, with no forcing term, is

$$V_{\mathrm{m}}(x) = \alpha\,\mathrm{e}^{-x/\lambda} + \beta\,\mathrm{e}^{x/\lambda}, \tag{3.20}$$

where α and β are constants that depend on the initial and boundary conditions, and where λ is found to be

$$\lambda = \sqrt{\frac{2\,R_{\mathrm{m}}}{a\,R_{\mathrm{l}}}} = \sqrt{\frac{r_{\mathrm{m}}}{r_{\mathrm{l}}}}. \tag{3.21}$$

If we impose the constraint that

$$\lim_{x\to\infty} \left|V_{\mathrm{m}}(x)\right| < \infty, \tag{3.22}$$

then we find that $\beta = 0$, and the voltage must fall exponentially with distance from the point of injection, at a rate λ, starting at a point $\alpha = V(0)$. Knowing the potential at any point on the cable is sufficient to solve for α : using Ohm's law, the voltage at $x = 0$, where I_{app} is being injected, is $\sigma\, r_{\mathrm{m}}$. Hence,

$$V_{\mathrm{m}}(x) = \sigma\, r_{\mathrm{m}}\, \mathrm{e}^{-|x|/\lambda}, \tag{3.23}$$

where $V_0 = V(0) = \sigma\, r_{\mathrm{m}}$ is frequently used to refer to the scale component of the solution.

The quantity λ is known as the characteristic lengthscale, and describes how rapidly the exponentially-decaying voltage drops as a function of space. A large lengthscale λ implies that current is able to diffuse long distances, while a small λ means that the voltage decays quickly as we measure further from the point of current injection. A set of example solutions (3.23) is shown in Figure 3.1 for different lengthscales. "Typical" values of λ do not exist per se, due to the high variation in morphologies. With realistic values for the physical parameters, however, λ tends to be in the region of 1 to 10 mm [Tuckwell, 1988].

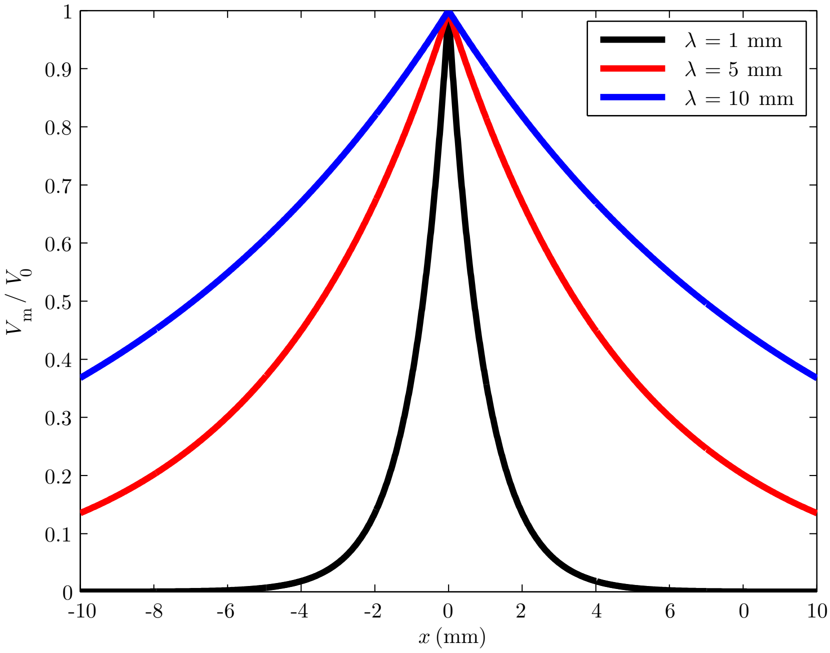


Figure 3.1: The steady-state voltage on an infinite cylinder, with constant current $I_{\mathrm{app}} = \sigma\,\delta(x)$ injected at $x = 0$, for three different values of the space constant, λ.

This length constant allows us to introduce a dimensionless, *electrotonic* distance, $X = x/\lambda$, and hence to consider the length l of a neuronal cable in dimensionless form using its electronic length, $L = l/\lambda$. These dimensionless properties allow different neuronal structures to be compared more readily, without having to consider radii and cytosolic resistances. We can then compare the electrotonic behaviour of different neurons by using the same language : the total electrotonic length of a dendritic tree is a measure of how "electrically compact" the tree is [Zador *et al.*, 1995]. Therefore, whilst neurons may vary wildly in size, we can compare their spatial electrotonic properties, or how much they will attenuate a current injected some distance away, more readily from their characteristic lengthscale, λ.

Timescale τ

The time constant τ (measured in ms) is similar : it describes how rapidly the transmembrane voltage V_m decays back to its resting state $V_\mathrm{m} = 0$ from any non-zero value. Let us then consider a patch of dendritic membrane satisfying

$$\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}t} = -\frac{1}{R_\mathrm{m}\, C_\mathrm{m}}\, V_\mathrm{m}. \tag{3.24}$$

If this patch of membrane is held at a value of $V_\mathrm{m} = V_0 > 0$, and, at time $t = 0$, we release the voltage clamp holding the membrane's potential at this level, then the membrane will relax to zero according to

$$V_\mathrm{m}(t) = V_0\, \mathrm{e}^{-\frac{t}{R_\mathrm{m}\, C_\mathrm{m}}}. \tag{3.25}$$

This exponential decay has a constant of $\tau = R_\mathrm{m}\, C_\mathrm{m}$, such that the voltage will have decayed to V_0/e by time $t = \tau$. This fundamental property of the system can be used to measure a dendritic tree's responsiveness : trees with large τ have a membrane voltage that responds slowly to stimulus, while those with small τ react rapidly to the effects of injected current, rising and falling quickly in response to the stimuli. A typical value for the membrane time constant is $\tau = 20$ ms; the time constant can range from $\tau = 1$ ms for neurons that encode fine-grained temporal information, to $\tau = 100$ ms for some hippocampal neurons [Spruston and Johnston, 1992]. As we can see here, the membrane time constant τ is a function of only transmembrane properties, namely the resistivity and capacitance.

Figure 3.2 shows how a patch of membrane responds to a discontinuous change in the transmembrane potential at time $t = 0$, for different values of τ, and Figure 3.3 shows an example voltage trace for an isolated patch of membrane being stimulated by delta spikes and a square pulse current, with the stimulus shown below.

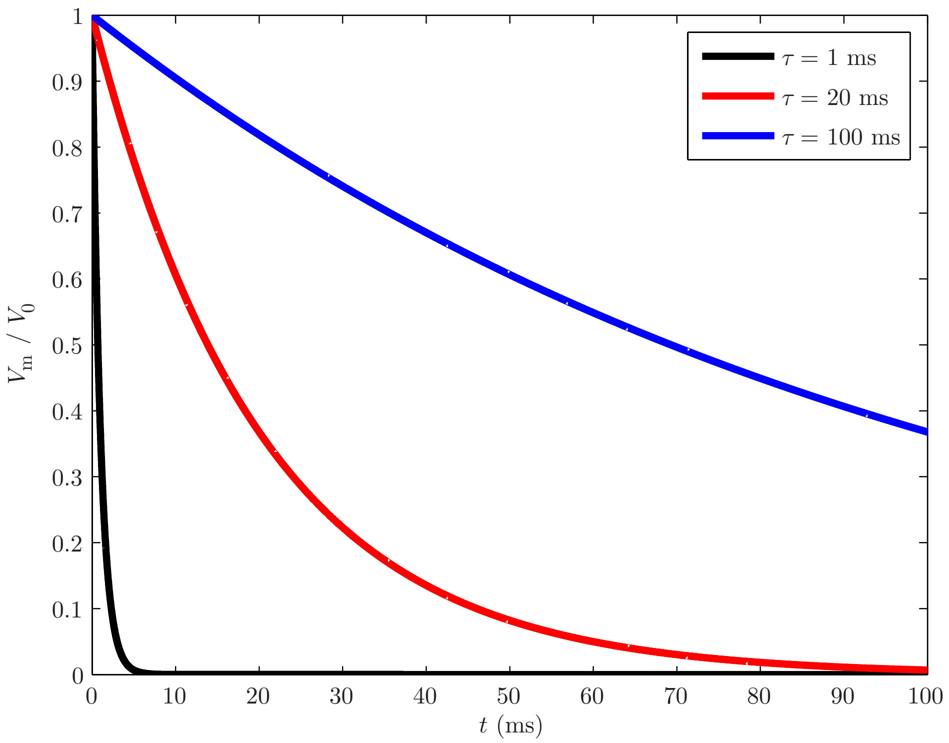


Figure 3.2: The transmembrane potential's response to the release of a voltage clamp. The voltage had been held at V_0, and the clamp was released at $t = 0$, at which point the potential relaxes to zero as in (3.25), for different values of the membrane time constant, τ.

3.1.4 Assumptions

Some of the key assumptions of dendritic cable theory stem from core conductor theory : the one-dimensional aspect of the equation is cable theory's most simplifying assumption, allowing any radial dependence of V_m to be neglected [Rall, 1977]. A wide body of evidence, both experimental [Rashbass and Rushton, 1949; Taylor, 1963; Cole, 1968] and theoretical [Plonsey, 1964; Clark and Plonsey, 1968; Rall, 1969; Eisenberg and Johnson, 1970], suggests that the errors generated in making this assumption are negligible.

The cytosolic core of the cable is assumed to provide a simple ohmic resistance to current flow. This, together with the assumption of the dependence of voltage on only one spatial dimension, allows the intracellular space to be characterised by R_l, the specific longitudinal resistivity, a parameter that does not depend on space. This, however, does not imply that the cable must have uniform resistance – as we will discover in the next chapter, it is possible to allow the radius of the cables to

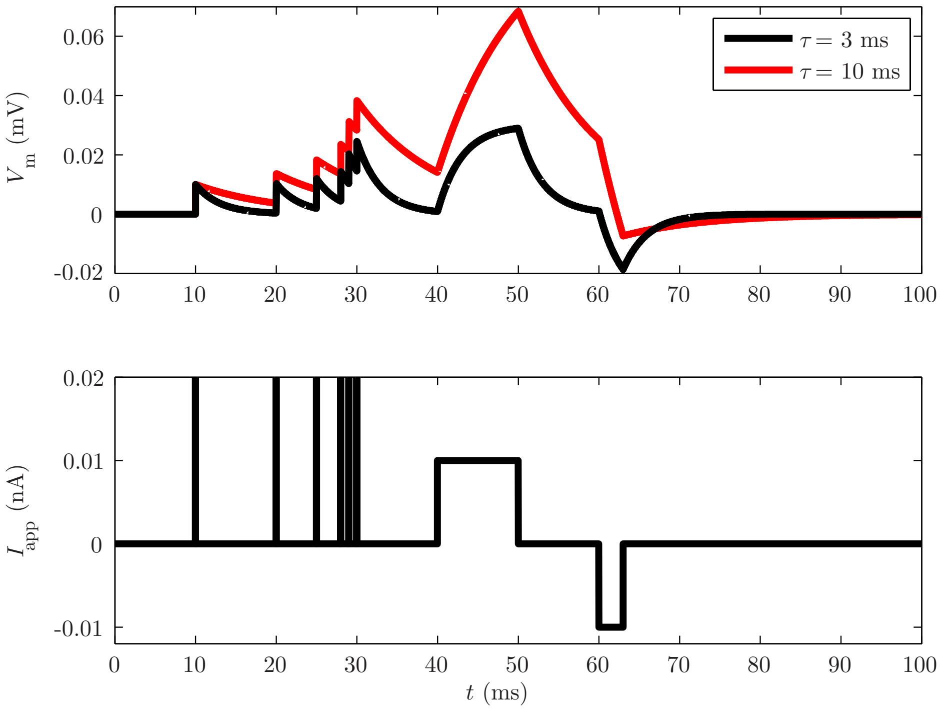


Figure 3.3: Perturbation of the transmembrane potential by injected current I_app, a train of delta spikes with integral 0.01, and two square pulses of magnitude 0.01, for two different membrane time constants. It is clear that membranes with a larger time constant take longer to adapt to perturbations and, as a result, may become more depolarised for an equivalent positive stimulus than a membrane that relaxes more quickly to its steady state. These voltage traces were obtained by numerical computation. These graphs were obtained by numerical simulation.

vary in a discontinuous sense, in order to approximate a tapering in the dendrites.

A related assumption is that both the intracellular and extracellular spaces are homogeneous. In reality, the cytosol contains a plethora of organelles (such as the endoplasmic reticulum), as well as membranes and vesicles. Similarly, the extracellular medium is packed full of neighbouring neurons and glial cells. Fortunately, the characteristic lengthscale of the electrical dynamics, λ, is several orders of magnitude greater than the scale of these heterogeneities, rendering this assumption valid for neural cells.

Further to this, we assume that any charge in the extracellular or intracellular fluids dissipates extremely rapidly. Any capacitive or inductive effects in these fluids can therefore be safely neglected [Scott, 1971]. In addition, the vast majority of current in a cable flows longitudinally, due to the large transmembrane resistivity, R_m, and because cables have radii much smaller than their lengths. Therefore, only a small amount of current escapes the cable into the large extracellular space. It is hence safe to assume that no voltage gradients exist in the extracellular space – they are simply too small and would diffuse too rapidly to affect the dynamics of the system. The extracellular fluid is thus modelled at isopotentiality. Because the amount of current entering the extracellular space from the cables is very small, the fluid is assumed to have a potential of $V_{\mathrm{e}} = 0$.

When current flows in cable-like structures, they induce a magnetic field. These are assumed to be negligible, an assumption backed in work by Rosenfalck [1969]; the magnetic fields are shown to be extremely small due to the very low magnitude of currents flowing through the cable.

Electrical current is, in physical form, the flow of a large number of ions. Cable theory assumes that these ions are both sufficiently numerous that the voltage may take continuous values, and that they are never depleted. Even in the very small (20 nm) space between chemical synapses, there are 36,000 potassium ions to be found, with all other ions in greater concentrations [Holt, 1998]. Even with constant electrical activity, Qian and Sejnowski [1989, 1990] found that ionic concentrations do not vary siginificantly, except in very small structures, such as in dendritic spines.

Except in rare cases, these assumptions prove to be well-founded for the system being modelled : a linear, passive dendritic cable, with the potential never deviating too far from resting potential. With currents injected by electrode rather than by changes in synaptic conductivity, closed-form solutions can be found; changes in conductivity due to synaptic stimuli are analytically more difficult to treat. For the modelling of subthreshold currents in dendritic trees, where stimulation occurs via injection of current rather than synaptic input, this framework is therefore a good approximation, and offers a valuable compromise between analytical tractability and computational efficiency.

3.2 A Note on Integral Transforms

The Laplace Transform

The Laplace transform, denoted by the linear operator $\mathcal{L}$, is defined as

$$\begin{aligned} \bar{f}(s) &= \mathcal{L}\left[\, f \,\right](s) \\ &= \int\limits_0^\infty f(t)\, \mathrm{e}^{-st}\, \mathrm{d}t, \end{aligned} \tag{3.26}$$

where the parameter $s = \sigma + i\omega$ is a complex number, with $\sigma,\, \omega \in \mathbb{R}$. For the Laplace transform $\bar{f}(s)$ of a real-valued function $f(t)$ to exist, $f(t)$ must be locally integrable on the interval $[0, \infty)$ and the integral (3.26) must converge. This holds true if $f(t)$ is of exponential type : there must exist $a,\, b \geq 0$ such that $\big|f(t)\big| \leq a\,\mathrm{e}^{bt}$ for all $t \geq 0$.

Analytically, the Laplace transform turns differential equations into polynomials, making them vastly easier to solve. For simple problems, the Laplace transform can be found by inspection, using tables that match an expression in the time domain with its analogue in the Laplace domain. This is the case for the cable equation for the infinite cable (2.11). Knowing that $\mathcal{L}[f'(t)] = s\bar{f}(s) - f(0)$, where $f'(t)$ is the first derivative of $f(t)$, then we can express the cable equation (3.1) in the Laplace domain as

$$s\,\bar{V}_{\mathrm{m}}(x,s) - V_{\mathrm{m}}(x,0) = \frac{\lambda^2}{\tau}\,\frac{\partial^2 \bar{V}_{\mathrm{m}}(x,s)}{\partial x^2} - \frac{\bar{V}_{\mathrm{m}}(x,s)}{\tau}. \tag{3.27}$$

If we introduce $\gamma^2(s) = \dfrac{\tau s + 1}{\lambda^2}$, then this becomes

$$\gamma^2(s)\,\bar{V}_{\mathrm{m}} = \frac{\partial^2 \bar{V}_{\mathrm{m}}}{\partial x^2} \tag{3.28}$$

assuming the zero initial condition, $V_{\mathrm{m}}(x,0) = 0$. Then, (3.28) is satisfied by

$$\bar{V}_{\mathrm{m}}(x,s) = \alpha(s)\,\mathrm{e}^{-\gamma(s)\,x} + \beta(s)\,\mathrm{e}^{\gamma(s)\,x}. \tag{3.29}$$

Once the resulting algebraic problem is solved in the Laplace domain, the solution is inverted back to its original domain using the inverse Laplace transform. For the case of cable problems, we typically apply the solution (3.29) to a tree with the relevant boundary conditions at branching points, at which point the system of equations becomes too complex to invert back into the time domain using tables. For systems of modest complexity, it may be possible to simply evaluate the inverse Laplace transform analytically, as defined by the Bromwich integral :

$$\begin{aligned} f(t) &= \mathcal{L}^{-1}\big[\,\bar{f}\,\big](t) \\ &= \lim_{T\to\infty} \frac{1}{2\pi i} \int\limits_{\gamma - iT}^{\gamma + iT} \bar{f}(s)\,\mathrm{e}^{st}\,\mathrm{d}s. \end{aligned} \tag{3.30}$$

This contour integral is performed along the vertical line $Re(s) = \gamma$ in the complex plane. Here, γ must be to the right of any poles, to ensure the contour path is in the integral's region of convergence. For isolated singularities, this is the standard approach. However, due to the multiplication by an exponential function of time, the inverse Laplace transform is an inherently sensitive and ill-posed problem. Any errors become exponentially divergent, an issue partially-resolved by using multiple precision arithmetic computational libraries, which lead to heightened accuracy but extremely slow evaluation of the integral. A number of Laplace transform inversion algorithms have been put forth, each of which demonstrates strengths for only a particular type of function. There exist methods based on evaluating an arbitrary number of derivatives, based on Post's Formula [Post, 1930; Gaver, 1966]; on Laguerre polynomial expansions [Weeks, 1966]; on Fourier series expansions [de Hoog *et al.*, 1982]; and on evaluating contours that deform around the singularities [Talbot, 1979]. Algorithms based on improvements or accelerations to these methods are still the subject of research today.

The Fourier Transform

The Fourier transform is closely-related to the Laplace transform. It also maps a function of space or time into an analogous frequency domain, where units are in cycles per second. Unlike the Laplace transform, however, for the Fourier transform $\hat{f}(\omega)$ of a real-valued function $f(x)$, the frequency argument ω is real.

The unitary, angular form of the Fourier transform is defined as

$$\begin{aligned}\hat{f}(\omega) &= \mathcal{F}\big[\,f\,\big](\omega)\\ &= \frac{1}{\sqrt{2\pi}}\int\limits_{-\infty}^{\infty} f(x)\,\mathrm{e}^{-i\omega x}\,\mathrm{d}x,\end{aligned}\tag{3.31}$$

and its inverse is

$$\begin{aligned}f(x) &= \mathcal{F}^{-1}\big[\,\hat{f}\,\big](x)\\ &= \frac{1}{\sqrt{2\pi}}\int\limits_{-\infty}^{\infty} \hat{f}(\omega)\,\mathrm{e}^{i\omega x}\,\mathrm{d}\omega.\end{aligned}\tag{3.32}$$

The Fourier transform $\hat{f}(\omega)$ of a function $f(x)$ exists provided that

$$\int\limits_{-\infty}^{\infty} \big|f(x)\big|\,\mathrm{d}x < \infty \tag{3.33}$$

and that $f(x)$ is a Lipschitz continuous function, satisfying

$$\big|f(x) - f(x')\big| \leq B\big|x - x'\big|^{\beta} \tag{3.34}$$

for any points x and x', for constant B and for $0 < \beta \leq 1$.

One important difference between the Fourier transform and the Laplace transform is the domain of integration. The traditionally-defined Laplace transform is unilateral, integrating over the non-negative reals (although a bilateral definition exists), whereas the Fourier transform is always two-sided, integrating over $(-\infty, \infty)$. For variables existing in a doubly-infinite domain, such as the spatial variable on an infinite cable, the Fourier transform can be the logical choice in terms of integral transforms.

Convolution Theorem

A great computational tool is the convolution theorem, which states that the Fourier transform of a convolution of two functions $f(t)$ and $g(t)$ is equal to the point-wise multiplication of their frequency-transform representations, $\hat{f}(s)$ and $\hat{g}(s)$. That is,

$$\begin{aligned} \mathcal{F}\Big[f \circledast g\Big](s) &= \mathcal{F}\big[\,f\,\big](s) \cdot \mathcal{F}\big[\,g\,\big](s) \\ &= \hat{f}(s) \cdot \hat{g}(s), \end{aligned} \tag{3.35}$$

where

$$\begin{aligned} \Big(f \circledast g\Big)(t) &= \int\limits_{0}^{\infty} f(t')\,g(t-t')\,\mathrm{d}t' \\ &= \int\limits_{0}^{\infty} f(t-t')\,g(t')\,\mathrm{d}t'. \end{aligned} \tag{3.36}$$

In terms of computational efficiency, a convolution done by point-wise multiplication in the Fourier domain is better than a convolution performed directly in the time domain : a simple time-domain convolution is a quadratic operation, requiring N^2 operations to obtain a result on functions of discrete length N, while a Fourier transform can be taken using the Fast Fourier Transform (FFT) [Cooley and Tukey, 1965] in quasilinear time, requiring $N \log N$ operations. The pointwise multiplication then requires N operations, and an inverse Fast Fourier Transform (iFFT) requires another $N \log N$, meaning that the convolution

$$\Big(f \circledast g\Big)(t) = \mathcal{F}^{-1}\Big[\mathcal{F}\big[\,f\,\big] \cdot \mathcal{F}\big[\,g\,\big]\Big](t) \tag{3.37}$$

is less than quadratic when performed as a frequency domain calculation, rather than directly in the time domain.

Integral transforms and convolutions are often used in signal processing and the analysis of linear systems, where they are of paramount importance to finding solutions to the systems' dynamics.

3.3 Some Concepts in Linear Systems Theory

Systems can be seen as "black box" systems where, if stimulated by some input, they respond in a way that is described by an operator, which maps a time-varying input, $x(t)$, to an output, $f(t)$. If we denote the operator of our hypothetical system by H, then we can describe the input-output relation of this system by

$$f(t) = H\big[\,x\,\big](t). \tag{3.38}$$

The cable equation (3.1) is such a system : were we to stimulate a cable at its resting potential by the injection of a current, it would respond in some characteristic manner. The cable equation can be classified as a second-order, linear, partial differential equation. Several of the approaches used to solve it will be specific to at least part of this description. For example, the cable equation's linearity ensures we can construct a solution for any number of inputs by simply solving it once for each input, and summing up the solutions. Before we review the literature for methods used to solve the cable equation analytically, it would be advantageous to discuss certain concepts in the theory of linear dynamical systems, which will aid us in developing algorithms for solving the equation on arbitrary trees.

Linearity

Systems that satisfy both the principles of superposition and scaling are known as linear systems. Superposition implies that, if a system responds to an input x with the output $f(x)$, then, for any combination of inputs $x_1, \ldots, x_n$, the following holds true :

$$f(x_1 + \ldots + x_n) = f(x_1) + \ldots + f(x_n). \tag{3.39}$$

The scaling property can be derived from superposition. If the function receives an input x an integer a number of times, then its output must scale accordingly :

$$f(ax) = af(x). \tag{3.40}$$

We can therefore consider any system, taking inputs $x_1(t)$ and $x_2(t)$ to produce the outputs $f_1(t) = H\big[x_1(t)\big]$ and $f_2(t) = H\big[x_2(t)\big]$, as linear, if and only if

$$a f_1(t) + b f_2(t) = H\big[\, a x_1 + b x_2 \,\big](t). \tag{3.41}$$

Time-Invariance

A system can be described as time-invariant if it obeys the time-shift property : its output does not explicitly depend on time. For example, if a given input $x(t)$ at time t produces an output $f(t) = H\big[\, x \,\big](t)$, then the same input applied at the delayed time $t + \Delta t$ will produce the same output, delayed by the same amount of time, such that $f(t + \Delta t) = H\big[\, x \,\big](t + \Delta t)$.

Impulse Response Functions

Any linear, time-invariant system can be fully characterised by its impulse response function, $h(t)$. This function is defined as the linear system's output when it is presented with a unit impulse :

$$h(t) = H\big[\, \delta \,\big](t), \tag{3.42}$$

where $\delta(t)$ is the Dirac delta function, a function with value zero everywhere except at $t = 0$, and with an integral of one, and where, as before, H is a linear operator. The impulse response function is of fundamental importance, because it allows us to compute the system's response, $f(t)$, as a function of any input, $x(t)$, by the convolution identity,

$$\begin{aligned} f(t) &= H\big[\, x \,\big](t) \\ &= \Big(x \circledast h\Big)(t) \\ &= \int\limits_0^\infty x(t')\, h(t - t')\, \mathrm{d}t'. \end{aligned} \tag{3.43}$$

The impulse response function's Laplace-domain analogue, $\bar{h}(s) = \mathcal{L}[\, h \,](t)$, is known as the transfer function. This frequency-domain representation has some significant advantages over the time-domain impulse response function. Being a frequency-domain function, it allows us to immediately determine the system's attenuation of certain frequencies; those that quench high frequencies while allowing low frequencies to pass are termed *lowpass* filters, while the opposite are *highpass* filters. A combination of these can create a *bandpass* filter, where only a range of frequencies are passed and all others are attenuated. Such information can help us predict the system's response to various stimuli. In addition, by convolution theorem, we can evaluate a convolution using the transfer function :

$$\Big(x \circledast h\Big)(t) = \mathcal{L}^{-1}\Big[\big(\mathcal{L}\big[\,x\,\big]\big) \cdot \bar{h}\Big](t). \tag{3.44}$$

Note that, in (3.37), we saw that a convolution could be written as the inverse Fourier transform of the pointwise product of the Fourier transforms of two functions. Here, we have used the Laplace transform as our frequency-domain integral transformation; more commonly, however, the Fourier transform is used, with convolution theorem holding for a range of integral transforms, of which the Fourier, Laplace, and bilateral Laplace transforms. This is especially important in computational mathematics, where convolutions are almost exclusively performed in the Fourier domain, thanks to the extremely efficient FFT algorithm. In the majority of approaches to solving cable problems described in Section 2.7.1, the resulting solution is found in the Laplace domain, meaning we can readily obtain $\bar{h}(s)$. For certain well-behaved systems, with no singularities in the right half of the complex plane, the Laplace-domain solution is analogous to a Fourier-domain function, after a change of variables. In theory, therefore, we could obtain the time-domain response of the dendritic system, $f(t)$, by multiplying the transfer function, $\bar{h}(s)$ with the Laplace transform of our input function, $\mathcal{L}\big[\,x\,\big](s)$, and taking an inverse Fourier transform after the change of variables, or by inverting the transfer function from the Laplace domain into the time domain, and then taking a convolution with our input, $x(t)$, using the Fast Fourier Transform.

Green's Functions

There are numerous approaches to solving differential equations, each of which may be amenable to certain types of equations. The Green's function approach is a powerful method of obtaining a fundamental solution to inhomogeneous differential equations, and is of central importance to this work. In order to set the context for this approach, we begin with a reminder of the concepts of homogeneity, and how differential equations can be described by their operators.

An n^{th}-order ordinary differential equation has the general form,

$$\sum_{k=0}^{n} p_k(t)\,\frac{d^k y}{dt^k} = q(t). \tag{3.45}$$

If $q(t)$ is non-zero, then it is called the *source* or *forcing* term, and the equation is *inhomogeneous.* This equation is also linear, because any function of $y(t)$ or its derivatives are linear. Linearity ensures that we can write this equation in the form,

$$L\big[\,y\,\big](t) = q(t), \tag{3.46}$$

where L is the linear differential operator,

$$L = \sum_{k=0}^{n} p_k(t)\,\frac{d^k}{dt^k}. \tag{3.47}$$

This is an example of an ordinary differential equation, but this holds for partial differential equations such as the cable equation. For the case of equation (3.1), the linear differential operator can be written

$$L = \frac{\partial}{\partial t} - \frac{\lambda^2}{\tau}\frac{\partial^2}{\partial x^2} + \frac{1}{\tau}, \qquad t \geq 0, \quad -\infty < x < \infty, \tag{3.48}$$

leaving the source term $q(x,t) = 0$ for this homogeneous partial differential equation. When we were to include a source term $I_{\text{app}}(x,t)$, simulating a current being applied to the cable at some point x at time t, then we would set $q(x,t) = I_{\text{app}}(x,t)$ and we would have to solve for an inhomogeneous system instead.

This is where the Green's function becomes a useful tool. Referring back to our generic ordinary differential equation (3.45), then the Green's function for this equation, with linear differential operator (3.47), is defined as the function $G(t,t')$ such that

$$L\big[\,G(\,\cdot\,,t')\,\big](t) = \delta(t - t'). \tag{3.49}$$

If we then multiply with the source term $q(t')$ and integrate, we obtain

$$\begin{aligned} \int_0^\infty L\big[\,G(\,\cdot\,,t')\,\big](t)\,q(t')\,\mathrm{d}t' &= \int_0^\infty \delta(t-t')\,q(t')\,\mathrm{d}t' \\ &= q(t), \end{aligned} \tag{3.50}$$

which is equal to $L\big[y(t)\big]$, as in (3.46). Because L is a differential operator acting only on t and not the integration variable t', it can be taken out of the integral to give

$$\begin{aligned} L\left[\int_0^\infty G(\,\cdot\,,t')\,q(t')\,\mathrm{d}t'\right](t) &= q(t) \\ &= L\big[\,y\,\big](t), \end{aligned} \tag{3.51}$$

and can then be inverted such that

$$y(t) = \int_0^\infty G(t,t')\,q(t')\,\mathrm{d}t'. \tag{3.52}$$

Therefore, evaluating this integral will allow us to find the solution $y(t)$ to the inhomogeneous differential equation (3.45). This elegant result means that, for any type of perturbation or input $q(t)$ into the system (3.46), we can find the solution $y(t)$, assuming that we have the Green's function $G(t,t')$ for the differential operator L. Finding the Green's function can therefore become the focus of solving a system of equations. It can be found using Laplace transforms, the method of images, and eigenvalue expansions [Cole *et al.*, 2011].

As a final remark, we note that, by solving a linear system of the form $L\big[\,y\,\big](t) = q(t)$ using the Green's function $G(t,t')$, we have effectively found the system's inverse operator L^{-1} such that $y(t) = L^{-1}\big[\,q\,\big](t)$. This inverse operator is therefore represented by the integration kernel,

$$L^{-1}\big[\,q\,\big](t) = \int_0^\infty G(t,t')\,q(t')\,\mathrm{d}t'. \tag{3.53}$$

The Cable Equation as a Linear Dynamical System

Much of the description of the mathematical tools used to deal with linear dynamical systems was exemplified using the hypothetical systems, (3.38) and (3.47). These are, of course, not unrelated systems – both are linear dynamical systems being presented with some input, $x(t)$ or $q(t)$, and responding with $f(t)$ and $y(t)$ respectively. With respect to the Green's function for cable systems, we can immediately equate these, but to steer around any confusion, we will use the notation prevalent in cable theory. Let $I_{\mathrm{app}}(x,t)$ be the source term, equivalent to $x(t)$ or $q(t)$, and let $V_{\mathrm{m}}(x,t)$ be the response of the system, analogous to $f(t)$ and $y(t)$. We can therefore rewrite the system as

$$H\big[\,I_{\mathrm{app}}\,\big](x,t) = V_{\mathrm{m}}(x,t), \tag{3.54}$$

or

$$L\big[\,V_{\mathrm{m}}\,\big](x,t) = I_{\mathrm{app}}(x,t). \tag{3.55}$$

For this dynamical system, we have already written down the differential operator L describing the cable equation, in (3.48). This begs the question : how is H, the operator which maps the input $I_{\mathrm{app}}(x,t)$ to the system's output, $V_{\mathrm{m}}(x,t)$, related to the system's differential operator, L ? We can consider (3.54) to be a *forwards* problem, mapping an input to an output. In contrast, the system (3.55) is an *inverse* problem : we need to somehow invert the differential operator L and apply the inverse to the source term in order to obtain a solution for the quantity of interest, $V_{\mathrm{m}}(x, t)$.

Informally, the inverse of L is encoded in its Green's function, $G(x, x', t, t')$, as in (3.53). We have already seen that if we take the integral of the product of the Green's function with the source term, we are able to solve for the system's response :

$$V_{\mathrm{m}}(x, t) = \int\limits_{0}^{\infty} \int\limits_{-\infty}^{\infty} G(x, x', t, t') \, I_{\mathrm{app}}(x', t') \, \mathrm{d}x' \, \mathrm{d}t'. \tag{3.56}$$

We can consider a simple injection of a delta stimulus at some point y at time $t = 0$ as an example. Letting $I_{\mathrm{app}} = \delta(x - y) \, \delta(t)$, and solving (3.56) :

$$\begin{aligned} V_{\mathrm{m}}(x, t) &= \int\limits_{0}^{\infty} \int\limits_{-\infty}^{\infty} G(x, x', t, t') \, \delta(x' - y) \, \delta(t') \, \mathrm{d}x' \, \mathrm{d}t' \\ &= G(x, y, t, 0). \end{aligned} \tag{3.57}$$

The operator H, mapping input to output, can then be defined as a convolution with the Green's function :

$$H\big[\, I_{\mathrm{app}} \,\big](x, t) = \Big(G \circledast I_{\mathrm{app}}\Big)(x, t). \tag{3.58}$$

This immediately brings to light another relation : the system's impulse response function, defined as $h(t)$ in (3.42), is equivalent to the Green's function. In this sense, the Green's function for the operator L is the system's fundamental response to a delta spike perturbation; this arises naturally from our definition of the Green's function in (3.49). The focus of the methods presented below is on solving for the Green's function, in order to evaluate $V_{\mathrm{m}}(x, t)$ using the convolution integral (3.56).

3.4 Steady-State and Time-Dependent Solutions

The diffusion of current along a cable will be affected by any boundaries or terminals it encounters. Equation (3.23) represents the steady state solution to the cable equation on the infinite cable. In this section, we will derive solutions to the cable equation in different situations, such as the steady-state solutions on the finite cable and, of course, the general solution.

3.4.1 Boundary Conditions for the Single Cable

At terminal points, where the cable ends, several boundary conditions can be considered. If we assume the dendrite has been cut, such that the cytosol is in direct contact with the extracellular fluid at the terminal, then the appropriate boundary condition is

$$V_{\mathrm{m}}(x_{\mathrm{term}}, t) = 0. \tag{3.59}$$

This is typically referred to as a *killed* end or a *short-circuit* termination. More relevant to healthy dendrites is the *sealed* or *closed* boundary condition, where the dendritic terminal ends normally and the cytosol is never in direct contact with the extracellular medium. This imposes a zero-current condition at the terminal, such that

$$\left.\frac{\partial V_{\mathrm{m}}(x,t)}{\partial x}\right|_{x=x_{\mathrm{term}}} = 0. \tag{3.60}$$

It is possible to voltage-clamp the dendritic tip, such that

$$V_{\mathrm{m}}(x_{\mathrm{term}}, t) = V_c, \tag{3.61}$$

some constant potential at which the terminal is clamped. Finally, on a closed cable, we can inject a constant current of magnitude σ, through the membrane and directly into the cytosol using an electrode. The boundary condition at this point is then

$$\left.\frac{\partial V_{\mathrm{m}}(x,t)}{\partial x}\right|_{x=x_{\mathrm{term}}} = -r_{\mathrm{l}}\,\sigma. \tag{3.62}$$

from (3.10), and as in Tuckwell [1988].

3.4.2 Steady State on a Semi-Infinite Cable

We can solve the cable equation for the semi-infinite cable, where $x \in [0, \infty)$. The equation describing the dynamics of the voltage is a homogeneous differential equation of the form,

$$\lambda^2 \frac{\mathrm{d}^2 V_{\mathrm{m}}(x)}{\mathrm{d}x^2} - V_{\mathrm{m}}(x) = 0, \tag{3.63}$$

where λ is defined as in (3.21). As we noted in Section 3.1.3, the solution to this equation is

$$V_{\mathrm{m}}(x) = \alpha\, \mathrm{e}^{-x/\lambda} + \beta\, \mathrm{e}^{x/\lambda}. \tag{3.64}$$

At the terminal, we have a closed dendritic tip, through which we inject a constant current $I_{\mathrm{app}}(x) = \sigma\,\delta(x)$ until equilibrium is reached. The voltage at the terminal must then satisfy (3.62). We also require that the voltage remain finite as the cable goes to infinity, such that

$$\lim_{x\to\infty} \big|V_\mathrm{m}(x)\big| < \infty. \tag{3.65}$$

Substituting (3.64) into (3.65), we see that the coefficient $\beta = 0$. By differentiating (3.64) and substituting the result into (3.62), we can evaluate the remaining coefficient, α :

$$\begin{aligned} \left.\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}x}\right|_{x=0} &= -\frac{\alpha}{\lambda} \\ &= -r_\mathrm{l}\,\sigma. \end{aligned} \tag{3.66}$$

Thus, the steady-state voltage on a semi-infinite cable, when a constant current σ is injected into the terminal, is

$$V_\mathrm{m}(x) = \lambda\, r_\mathrm{l}\, \sigma\, \mathrm{e}^{-x/\lambda}, \tag{3.67}$$

and the voltage decays exponentially with distance from the terminal, with decay rate λ. The magnitude of the solution at any point in space is proportional to the magnitude of the current, σ, and to the longitudinal resistance per unit length, r_l. Example voltage solutions are shown in Figure 3.4.

3.4.3 Steady State on a Finite Cable with Closed Ends

We will consider a finite cable with sealed ends at $x = 0$ and $x = L$. Let us assume that the far side of our cable has Neumann boundary conditions : it is a closed, healthy piece of dendritic cable with no current leaking through the tip, satisfying (3.60). As above, we will consider the case where the near side of the cable, at $x = 0$, is being subjected to the injection of a constant current $I_\mathrm{app}(x) = \sigma\,\delta(x)$, such that the boundary satisfies (3.62). The distribution of voltage follows (3.64). We can take its derivative to find

$$\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}x} = \frac{\beta}{\lambda}\,\mathrm{e}^{x/\lambda} - \frac{\alpha}{\lambda}\,\mathrm{e}^{-x/\lambda}. \tag{3.68}$$

Then, substituting the boundary condition for the injected current at $x = 0$, we find

$$\left.\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}x}\right|_{x=0} = \frac{\beta-\alpha}{\lambda} = -r_\mathrm{l}\,\sigma. \tag{3.69}$$

From this, we obtain

$$\beta = \alpha - \lambda\, r_\mathrm{l}\, \sigma. \tag{3.70}$$

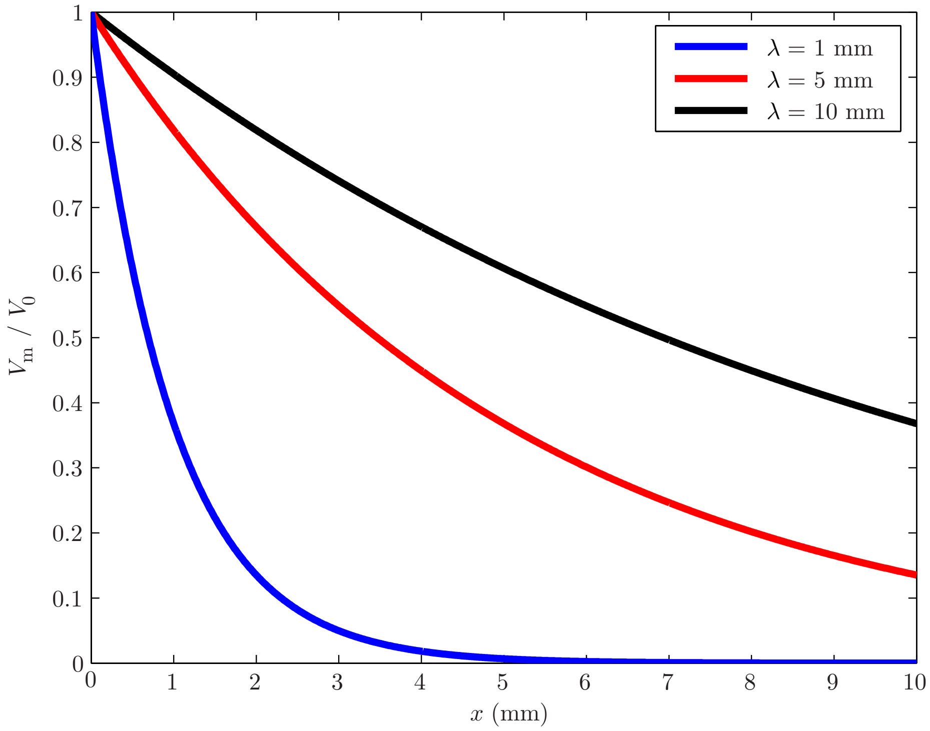


Figure 3.4: The steady-state voltage on a semi-infinite cylinder, with constant current σ injected at $x = 0$, for three different values of the space constant, λ.

We can do the same for the other boundary at $x = L$, substituting its zero-flow condition into (3.68) :

$$\left.\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}x}\right|_{x=L} = \frac{\beta}{\lambda}\,\mathrm{e}^{L/\lambda} - \frac{\alpha}{\lambda}\,\mathrm{e}^{-L/\lambda} = 0. \tag{3.71}$$

This yields

$$\beta = \alpha\,\mathrm{e}^{-2L/\lambda}. \tag{3.72}$$

Substituting (3.72) into (3.70), we get

$$\begin{aligned}\alpha\,\mathrm{e}^{-2L/\lambda} &= \alpha - \lambda\, r_{\mathrm{l}}\,\sigma \\ \alpha\big(1-\mathrm{e}^{-2L/\lambda}\big) &= \lambda\, r_{\mathrm{l}}\,\sigma \\ \alpha &= \frac{\lambda\, r_{\mathrm{l}}\,\sigma}{\big(1-\mathrm{e}^{-2L/\lambda}\big)} \qquad (3.73) \\ \beta &= \lambda\, r_{\mathrm{l}}\,\sigma\left(\frac{1}{\big(1-\mathrm{e}^{-2L/\lambda}\big)}-1\right)\end{aligned}$$

The steady-state solution for a cable with two closed ends, therefore, should we inject a current $I_{\mathrm{app}}(x) = \sigma\,\delta(x)$ at the terminal $x = 0$, is

$$V_{\mathrm{m}}(x) = \frac{\lambda\, r_{\mathrm{l}}\,\sigma}{\big(1-\mathrm{e}^{-2L/\lambda}\big)}\,\mathrm{e}^{-x/\lambda} + \lambda\, r_{\mathrm{l}}\,\sigma\left(\frac{1}{\big(1-\mathrm{e}^{-2L/\lambda}\big)}-1\right)\mathrm{e}^{x/\lambda}. \qquad (3.74)$$

This solution can be seen in Figure 3.5. As we expect, the solution scales linearly with the magnitude of the constant injected current, σ, as with the longitudinal resistance, r_{l}. The characteristic length λ continues to scale the rate at which the exponential decays with distance from the site of injection, but also the rate at which it rises as we near the other boundary, in addition to scaling the amplitude of the solution as with σ.

3.4.4 Steady State on a Finite Cable with One Open End

We can take the piece of dendritic cable from Section 3.4.3, and cut off the tip of the membrane at $x = L$, such that the cytosol is in direct contact with the extracellular potential. The boundary condition at this terminal is now

$$V_{\mathrm{m}}(L) = 0, \qquad (3.75)$$

with the potential difference across the membrane going to zero as the intracellular and extracellular media meet. The boundary condition at $x = 0$ remains the same as before : we inject a constant current $I_{\mathrm{app}}(x) = \sigma\,\delta(x)$ from time $t = 0$, wait until all transients have passed and the system reaches equilibrium. The current is injected at $x = 0$, so the boundary condition reads as in (3.62). The equation governing the distribution of current along the cable, as above, is (3.64). Taking

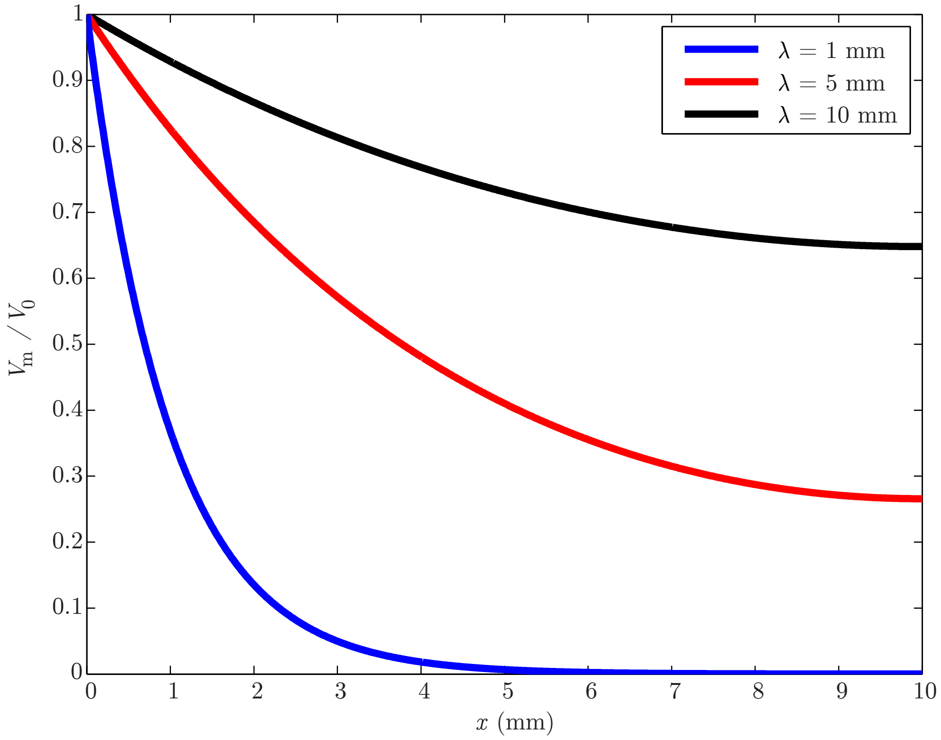


Figure 3.5: The steady-state solution on a finite cable with closed boundaries, and an injection at the terminal $x = 0$.

derivatives and substituting the open-end boundary condition :

$$V_{\mathrm{m}}(L) = \alpha\,\mathrm{e}^{-L/\lambda} + \beta\,\mathrm{e}^{L/\lambda} = 0, \tag{3.76}$$

we find that

$$\beta = -\alpha\,\mathrm{e}^{-2L/\lambda}. \tag{3.77}$$

The boundary condition at $x = 0$ remains the same as above, and so we find that

$$\beta = \alpha - \lambda\,r_1\,\sigma \tag{3.78}$$

as in (3.70). Substituting this into (3.77), we get

$$\begin{aligned}\alpha - \lambda\, r_\mathrm{l}\, \sigma &= -\alpha\, \mathrm{e}^{-2L/\lambda} \\ \alpha(1+\mathrm{e}^{-2L/\lambda}) &= \lambda\, r_\mathrm{l}\, \sigma \\ \alpha &= \frac{\lambda\, r_\mathrm{l}\, \sigma}{(1+\mathrm{e}^{-2L/\lambda})} \qquad (3.79) \\ \beta &= \lambda\, r_\mathrm{l}\, \sigma \left(\frac{1}{\left(1+\mathrm{e}^{-2L/\lambda}\right)} - 1 \right).\end{aligned}$$

The steady-state voltage distribution on a finite cable of length L, with an injected current of magnitude σ through a closed end at $x = 0$, with the distal terminal open to the cytosol, is

$$V_\mathrm{m}(x) = \frac{\lambda\, r_\mathrm{l}\, \sigma}{\left(1+\mathrm{e}^{-2L/\lambda}\right)}\, \mathrm{e}^{-x/\lambda} + \lambda\, r_\mathrm{l}\, \sigma \left(\frac{1}{\left(1+\mathrm{e}^{-2L/\lambda}\right)} - 1 \right) \mathrm{e}^{x/\lambda}. \qquad (3.80)$$

Figure 3.6 shows this solution for different values of the characteristic length constant, λ. This parameter plays an important role at governing the rate at which the voltage decays to zero as we approach the open terminal.

3.4.5 General Solution

The general, time-dependent solution to the inhomogeneous cable equation (where current is injected at any point along the cable, and is allowed to vary as a function of time) can be found readily by rewriting it in the frequency domain. We will derive the general solution to the cable equation in the form of the system's impulse response function. The cable equation for the infinite cable, as in (3.1), is

$$\frac{\partial V_\mathrm{m}(x,t)}{\partial t} = D \frac{\partial^2 V_\mathrm{m}(x,t)}{\partial x^2} - \frac{V_\mathrm{m}(x,t)}{\tau} + I_\mathrm{app}(x,t), \qquad (3.81)$$

with $-\infty < x < \infty$ and $t \geq 0$. Furthermore, we assume that initially, the cable is at rest such that

$$V_\mathrm{m}(x,0) = 0. \qquad (3.82)$$

Let us consider this system in the language of Green's functions. We have a linear differential operator, $L = L(x,t)$, defined similarly to as in (3.48), by

$$L = \frac{\partial}{\partial t} - D \frac{\partial^2}{\partial x^2} + \frac{1}{\tau}. \qquad (3.83)$$

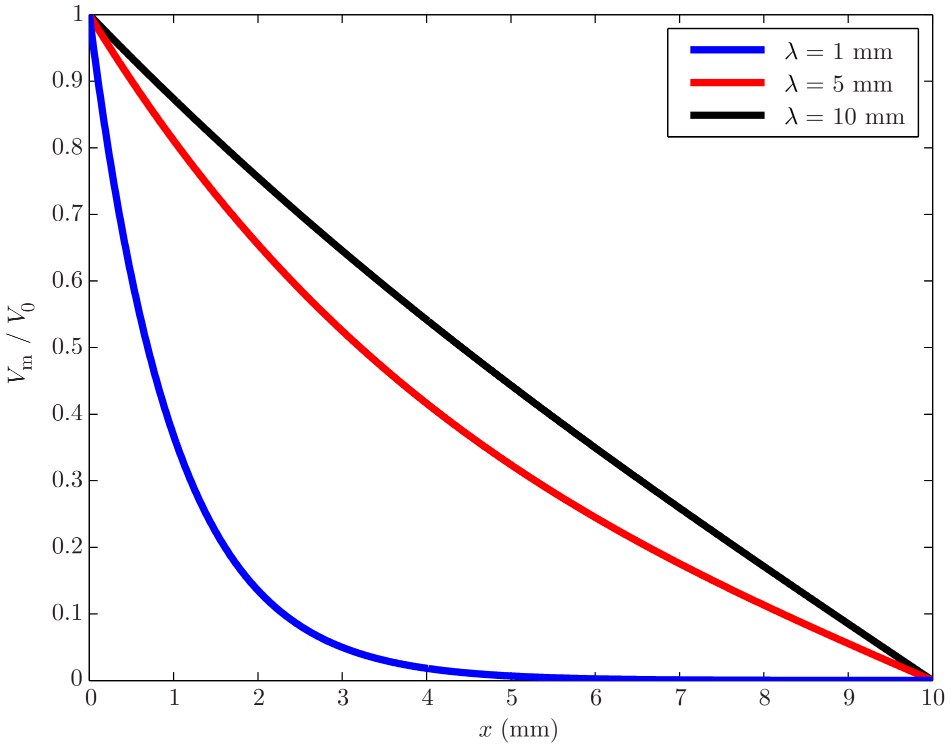


Figure 3.6: The steady-state solution on a finite cable with an open distal terminal tip, and an injection at the proximal, closed terminal.

When L acts on the function $V_{\mathrm{m}}(x,t)$, we see that

$$L\big[\,V_{\mathrm{m}}\,\big](x,t) = I_{\mathrm{app}}(x,t), \tag{3.84}$$

from (3.81). The Green's function for the operator L is a function which satisfies

$$L\Big[G(\,\cdot\,,x',\,\cdot\,,t')\Big](x,t) = \delta(x-x')\,\delta(t-t'). \tag{3.85}$$

If we multiply (3.85) by the injection term I_{app} and integrate, we find that

$$\begin{aligned}
&\int_0^\infty\int_{-\infty}^\infty L\Big[G(\,\cdot\,,x',\,\cdot t')\Big](x,t)\; I_{\mathrm{app}}(x',t')\,\mathrm{d}x'\,\mathrm{d}t' \\
&= \int_0^\infty\int_{-\infty}^\infty \delta(x-x')\,\delta(t-t')\,I_{\mathrm{app}}(x',t')\,\mathrm{d}x'\,\mathrm{d}t' \\
&= I_{\mathrm{app}}(x,t).
\end{aligned} \tag{3.86}$$

We then proceed as in Section 3.3, substituting (3.84) and (3.85) into (3.86) to give

$$\begin{aligned} L\Big[\,V_{\mathrm{m}}\Big](x,t) &= \int_{0}^{\infty}\int_{-\infty}^{\infty} L\Big[G(\,\cdot\,,x',\,\cdot\,,t')\Big](x,t)\; I_{\mathrm{app}}(x',t')\,\mathrm{d}x'\,\mathrm{d}t' \\ &= L\left[\int_{0}^{\infty}\int_{-\infty}^{\infty} G(\,\cdot\,,x',\,\cdot\,,t')\,I_{\mathrm{app}}(x',t')\,\mathrm{d}x'\,\mathrm{d}t'\right](x,t) \end{aligned} \tag{3.87}$$

and therefore,

$$V_{\mathrm{m}}(x,t) = \int_{0}^{\infty}\int_{-\infty}^{\infty} G(x,x',t,t')\,I_{\mathrm{app}}(x',t')\,\mathrm{d}x'\,\mathrm{d}t'. \tag{3.88}$$

As expected, this extremely useful result states that we can solve for the transmembrane potential for any input function I_{app} by simply taking the integral of a product of the Green's function, $G(x,x',t,t')$, with the injection term I_{app}. We must therefore solve (3.84) with I_{app} as a delta stimulus in both space and time in order to obtain a Green's function for the operator L.

Without loss of generality, we can consider $I_{\mathrm{app}}(x,t) = \delta(x-x')\,\delta(t)$, with current being injected at $t=0$ instead of at an arbitrary time t', and reintroduce this point later, by making use of the linear system's time-invariance. As such,

$$L\Big[G(\,\cdot\,,x',\,\cdot\,)\Big](x,t) = \delta(x-x')\,\delta(t). \tag{3.89}$$

Taking the Laplace transform (as discussed in Section 3.2) for the time variable, we find

$$\begin{gathered} \mathcal{L}\left[\frac{\partial G(x,x',\,\cdot\,)}{\partial t} - D\,\frac{\partial^2 G(x,x',\,\cdot\,)}{\partial x^2} + \frac{G(x,x',\,\cdot\,)}{\tau}\right](s) = \mathcal{L}\Big[\delta(x-x')\,\delta(\,\cdot\,)\Big](s) \\ s\,\bar{G}(x,x',s) - G(x,x',0) - D\,\frac{\partial^2 \bar{G}(x,x',s)}{\partial x^2} + \frac{\bar{G}(x,x',s)}{\tau} = \delta(x-x'). \end{gathered} \tag{3.90}$$

After substituting the initial condition $G(x,x',0)=0$, analogous to that in (3.82), this can be written

$$\gamma^2(s)\,\bar{G}(x,x',s) - D\,\frac{\partial^2 \bar{G}(x,x',s)}{\partial x^2} = \delta(x-x'), \tag{3.91}$$

where $\gamma^2(s) = s + 1/\tau$.

The Laplace transform allows us to write our differential equation in a simpler, more manageable form. This transform is particular well-suited to the time variable : time is defined as $t \geq 0$, and the traditional Laplace transform is a function of a real, non-negative argument. We wish to simplify the Laplace-domain cable equation (3.91), once again using an integral transform; however, the domain of integration must now be $(-\infty, \infty)$, because the spatial variable is defined along this domain. The Fourier transform, $\mathcal{F}$, is a sister transform to the Laplace transform, and has similar properties. Defined as in Section 3.2, it integrates the function over the domain we are interested in. We will apply the Fourier transform of (3.91) with respect to the spatial variable x, by inspection, and immediately solve the resulting algebraic equation for the Green's function :

$$\begin{aligned}\mathcal{F}\left[\gamma^2(s)\,G(\,\cdot\,,x',s) - D\,\frac{\partial^2 G(\,\cdot\,,x',s)}{\partial x^2}\right](\omega) &= \mathcal{F}\Big[\delta\big(\,\cdot\,-x'\big)\Big](\omega)\\ \gamma^2(s)\,\hat{\bar{G}}(\omega,x',s) + \omega^2 D\,\hat{\bar{G}}(\omega,x',s) &= \frac{\mathrm{e}^{i\omega x'}}{\sqrt{2\pi}} \\ \hat{\bar{G}}(\omega,x',s) &= \frac{1}{\sqrt{2\pi}D}\,\frac{\mathrm{e}^{i\omega x'}}{\big(\gamma^2(s)/D+\omega^2\big)}\end{aligned} \tag{3.92}$$

We can now invert the function back, first from the Fourier spatial frequency domain, back into standard space, and then from the unilateral Laplace temporal frequency domain, back into standard time. We therefore take an inverse Fourier transform, again by inspection. Using the fact that

$$\mathcal{F}^{-1}\left[\sqrt{\frac{2}{\pi}}\,\frac{1}{a^2+(\,\cdot\,)^2}\right](x) = \frac{\mathrm{e}^{-a|x|}}{a}, \tag{3.93}$$

we can take the inverse transform, using $a^2 = \gamma^2(s)/D$, such that

$$\begin{aligned}\mathcal{F}^{-1}\Big[\hat{\bar{G}}(\,\cdot\,,x',s)\Big](x) &= \mathcal{F}^{-1}\left[\frac{1}{2D}\sqrt{\frac{2}{\pi}}\,\frac{\mathrm{e}^{i(\,\cdot\,)x'}}{\big(\gamma^2(s)/D+(\,\cdot\,)^2\big)}\right](x)\\ &= \left(\frac{\mathrm{e}^{-\frac{\gamma(s)}{\sqrt{D}}|\,\cdot\,|}}{2\sqrt{D}\,\gamma(s)} \circledast \delta(\,\cdot\,-x')\right)(x),\end{aligned} \tag{3.94}$$

by convolution theorem. Because γ is a function of s, it makes sense to substitute these out for their functional forms and group terms in s for clarity. Thus,

$$\bar{G}(x,x',s) = \frac{1}{2\sqrt{D}}\left(\frac{\mathrm{e}^{-\frac{\sqrt{s+1/\tau}}{\sqrt{D}}|x-x'|}}{\sqrt{s+1/\tau}}\right). \tag{3.95}$$

This function is frequency-shifted : everywhere that s appears, there is a constant $1/\tau$ being added to it, shifting its frequency by $-1/\tau$. Knowing that

$$\bar{f}(s+a) = \mathcal{L}\Big[\, \mathrm{e}^{-a|\,\cdot\,|}\, f \,\Big](s), \tag{3.96}$$

for a function $f = f(t)$, we can make use of this shifting property by writing

$$\begin{aligned} \mathcal{L}^{-1}\Big[\, \bar{G}(x,x',\,\cdot\,) \,\Big](t) &= \mathcal{L}^{-1}\left[\frac{1}{2\sqrt{D}} \left(\frac{\mathrm{e}^{-\frac{\sqrt{(\,\cdot\,)+1/\tau}}{\sqrt{D}}|x-x'|}}{\sqrt{(\,\cdot\,)+1/\tau}} \right) \right](t) \\ &= \frac{\mathrm{e}^{-t/\tau}}{2\sqrt{D}}\, \mathcal{L}^{-1}\left[\frac{1}{\sqrt{(\,\cdot\,)}}\, \mathrm{e}^{-\frac{\sqrt{(\,\cdot\,)}}{\sqrt{D}}|x-x'|} \right](t) \\ &= \frac{\mathrm{e}^{-\frac{t}{\tau}}}{2\sqrt{D}}\, \frac{\mathrm{e}^{-\frac{(x-x')^2}{4Dt}}}{\sqrt{\pi t}}. \end{aligned} \tag{3.97}$$

As the final step of the derivation of this function, we will converge back onto the notation used in the literature. By denoting the Green's function for an infinite cable by a subscripted infinity, and replacing x' by the variable name y, we write and define the Green's function for the infinite cable as

$$G_\infty(x,y,t) := \frac{1}{\sqrt{4\pi D t}}\, \mathrm{e}^{-\frac{(x-y)^2}{4Dt}}\, \mathrm{e}^{-\frac{t}{\tau}}. \tag{3.98}$$

The function $G_\infty(x,y,t)$ is the system's impulse response function – the characteristic transmembrane response to a unit delta spike or perturbation, injected at $x = y$ and $t = 0$. In making use of it, we will typically assume that we inject an input current at time $t = 0$, although we could easily shift the response by some arbitrary amount t' by making use of the system's time-invariance property, and computing $G_\infty(x,y,t-t')$ instead.

The Green's function (3.98) retains the property that the transmembrane potential can be found for any generic current $I_{\mathrm{app}}(x,t)$ and given any arbitrary initial conditions $V_{\mathrm{m}}(x,0)$, by taking a product and integrating :

$$V_{\mathrm{m}}(x,t) = \int\limits_0^t \int\limits_{-\infty}^{\infty} G_\infty(x,y,t-t')\, I_{\mathrm{app}}(y,t')\, \mathrm{d}y\, \mathrm{d}t' + \int\limits_{-\infty}^{\infty} G_\infty(x,y,t)\, V_{\mathrm{m}}(y,0)\, \mathrm{d}y. \tag{3.99}$$

A space-time representation of the Green's function (3.98) is given in Figure 3.7. Figure 3.8 shows how the voltage along a cable dissipates, with an injection $I_{\mathrm{app}}(x,t)$ at $t = 0$ and point of injection $y = 5$ mm, for the initial condition $V_{\mathrm{m}}(x,0) = 0$.

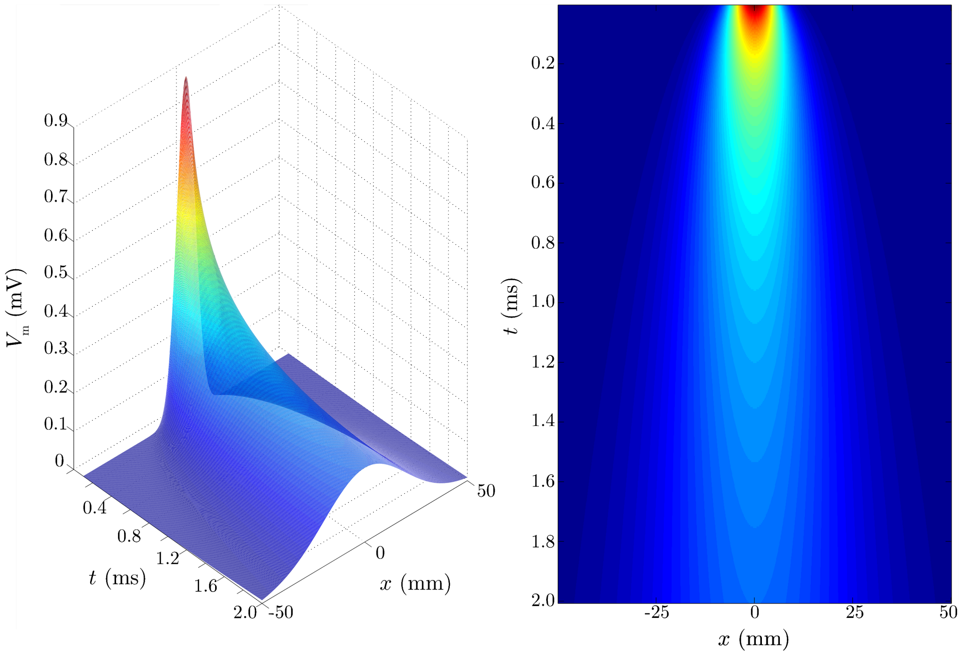


Figure 3.7: The Green's function, $G_\infty(x, y, t)$. Parameters were set at $\lambda = 10$ mm and $\tau = 100$ ms, similar to the space constant in a motorneuron. The time axis is truncated between $0 \leq t < 0.01$, as the function is extremely large in this interval (infinite at $t = 0$), making it very difficult to visualise.

3.4.6 Alpha Currents

The delta stimulus, a function that jumps to a value infinitely high in infinitessimal time, and yet with unit integral, is a highly unrealistic physiological perturbation. Its usefulness lies in its ability to provide us with a Green's function which, once calculated, can be used to solve for more biologically-realistic forms of current injection. Evidence [Jack and Redman, 1971] shows that synaptic input can be well-approximated using the alpha function,

$$\alpha(t) = b\, t\, \mathrm{e}^{-\frac{t}{T}}, \qquad t > 0, \tag{3.100}$$

where $T > 0$ is a scaling constant, determining the duration of the function. The parameter b can be positive or negative depending on whether the stimulus is excitatory or inhibitory. The alpha function, shown in Figure 3.9, grows to its maximum at $t = T$, and then relaxes to zero.

We have no closed-form solution for the transmembrane potential's response to stimulation by an alpha-like current. Tuckwell [1988] does provide two different

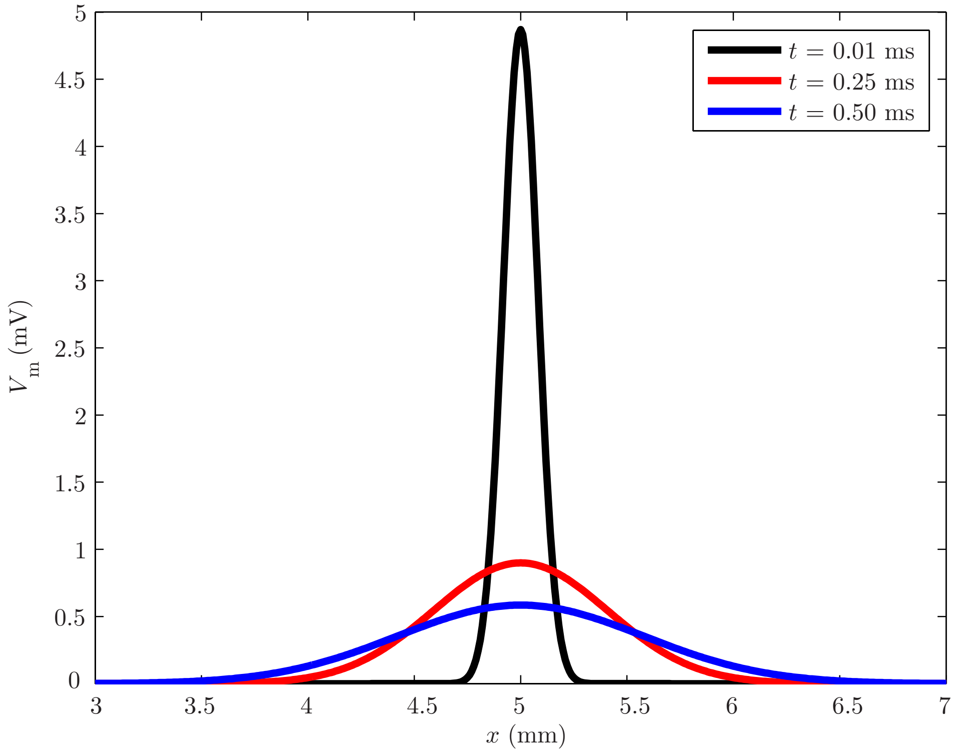


Figure 3.8: Voltage traces for injection at $y = 5$ mm on the infinite cable. Parameters were taken to be $\lambda = 1$ mm and $\tau = 3$ ms – a neuron with faster response than the one in Figure 3.7.

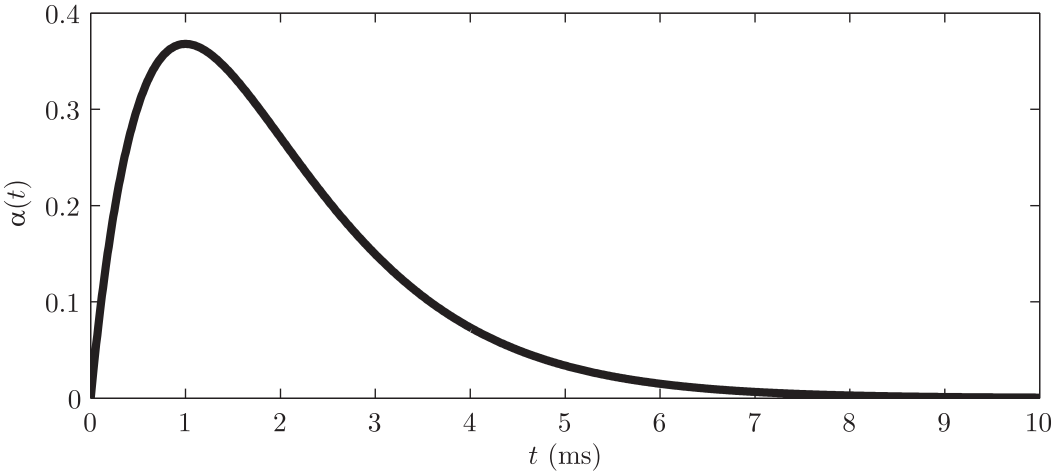


Figure 3.9: Alpha function $\alpha(t)$ with $T = 1$ ms.

expressions for $V_{\mathrm{m}}(x, t)$, for situations where $I_{\mathrm{app}} = \delta(x - y)\,\alpha(t)$, as calculated in (3.99) – one for short times, and one for long times, pointing out that both derivations make assumptions about the magnitude of T, and that, for the general case, a numerical evaluation of the integrals in (3.99) may be preferable.

3.4.7 Rectangular Pulse

We can, however, derive a closed-form solution for the cable equation for rectangular pulse currents : a constant current of magnitude σ, turned on instantaneously at time $t = 0$, and turned off after a duration d. For this form of current, we have

$$I_{\text{app}}(x,t) = \sigma\, \delta(x-y)\, \Theta(t)\, \Theta(d-t), \tag{3.101}$$

applied at y, where $\Theta(t)$ is the Heaviside step function :

$$\Theta(t) = \left\{ \begin{array}{ll} 0 & t < 0 \\ 1 & t \geq 0 \end{array} \right. . \tag{3.102}$$

The solution to (3.99), when $I_{\text{app}}(x,t)$ is an injected rectangular pulse of current, is given by Coombes and Bressloff [2003] as

$$V_{\text{m}}(x,t) = A(x, t - \min\{t, \tau_R\}) - A(x,t), \tag{3.103}$$

where $A(x,t)$ is given by

$$\begin{aligned} A(x,t) = \frac{\sigma}{4}\sqrt{\frac{\tau}{D}} \Bigg[& \exp\left(-\,|x|\sqrt{\frac{1}{D\,\tau}}\right) \text{erfc}\left(-\frac{|x|}{\sqrt{4Dt}} + \sqrt{\frac{t}{\tau}}\right) \\ & + \;\exp\left(|x|\sqrt{\frac{1}{D\,\tau}}\right) \text{erfc}\left(\frac{|x|}{\sqrt{4Dt}} + \sqrt{\frac{t}{\tau}}\right)\Bigg], \end{aligned} \tag{3.104}$$

and

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \mathrm{e}^{-x^2}\, \mathrm{d}x \tag{3.105}$$

is the complementary error function. The resulting transmembrane voltage response to this type of stimulus is shown in Figure 3.10.

3.5 The Path Integral for Dendritic Trees

The Green's function for the infinite cable is the fundamental building block for the construction of the time-domain solution to the cable equation on more complex spaces. Branching structures such as binary trees, representative of the morphol-

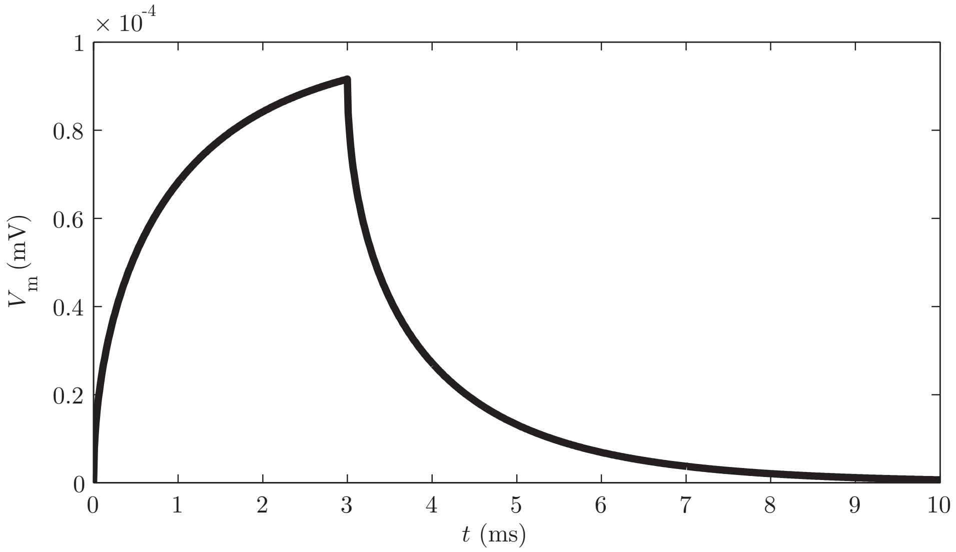


Figure 3.10: Transmembrane voltage response to a rectangular current, $I_\mathrm{app} = \sigma\,\delta(x)\,\Theta(t)\,\Theta(d-t)$ for a duration of $d = 3$ ms, amplitude $\sigma = 1$ nA, characteristic length constant $\lambda = 10$ mm and time constant $\tau = 2$ ms.

ogy of dendritic trees, can be used as the substrate for the diffusion of current in cable theory problems. Thus, by applying the appropriate boundary conditions at branching points, it is possible to solve for the Green's function on finite, branching trees.

At branching points, we apply Kirchhoff's circuit laws : continuity of the potential and a sum of zero net current. The prior imposes that, for a node where branch i at $x = 0$ meets branch j at $x = 0$, then

$$V_i(0,t) = V_j(0,t). \tag{3.106}$$

The latter states that

$$\sum_i \frac{1}{r_\mathrm{l}^i} \frac{\partial V_i}{\partial x}\bigg|_{x=0} = 0\,, \tag{3.107}$$

for all branches i having an x-coordinate of $x = 0$ at the node, and where r_l^i is the longitudinal resistance r_l for branch i. Terminals can have open or closed ends, as defined in Section 3.4.1, where closed tips, as in (3.60), are the most biologically-realistic.

For arbitrarily-branching trees, constructing a system of equations and boundary conditions quickly becomes very difficult. As demonstrated in the previous chapter, a great deal of work has gone into developing methods for finding a solution to the linear cable equation on complex branching structures. The majority of methods, however, are rooted in the Laplace domain, providing solutions which must be numerically inverted back into the time domain. With the definition of the inverse transform (3.30) containing an exponential function of time, numerical inverse Laplace transform methods are inherently extremely sensitive to parameters and to system precision, often requiring the use of multiple precision arithmetic libraries, which can lead to drastic losses in computational efficiency. The main alternatives are to use numerical simulation on a compartmentalised dendritic tree, based on Rall's compartmental model [Rall, 1964]; this remains the tool of choice for the vast majority of cable theory problems today, due to its simplicity and general applicability.

Numerical simulations can be computationally expensive, however, and only provide a discrete approximation to the continuous cable equation. With the availability of thousands of neuronal geometries in the form of standardised `.swc` geometry files obtainable from the NeuroMorpho [Ascoli *et al.*, 2007] database, representing over two hundred thousand manual reconstruction hours, a method for efficiently constructing the continuous Green's function solution, to high accuracy and in the time domain, would be extremely valuable. Abbott *et al.*'s [1991] path integral for dendritic trees provides this tool, allowing the Green's function to be written as an infinite, convergent sum of functions of trips on the tree.

Abbott *et al.*'s [1991] Path Integral

The Green's function, the response of a tree's transmembrane voltage to a unit pulse stimulus, captures the full effect of the tree's spatial extent. With a strong body of evidence demonstrating that the tree's complete morphology is essential to understanding how current diffuses along the branching structure, the ability to compress the entire tree into a single function is extremely beneficial. In order to compute the Green's function, therefore, we must sample from the entirety of the dendritic tree. We can capture the full detail of the dendritic morphology using a path integral for the Green's function.

The Feynman path integral stems from quantum mechanics, where it formulates a description of a system's trajectory between any two points, in terms of an integral, or sum, over all possible trajectories between these two points. For a particle moving in space, for example, there are an infinite number of trajectories from some point x in space, to a point y, each weighted in a probabilistic fashion. Assuming that the integral converges, it is possible to compute a property of the particle's trajectory (such as, in quantum mechanics, its quantum amplitude), by integrating over the infinite set of trajectories that the particle can take.

In 1991, Abbott *et al.* [1991] derived a path integral formalism for dendritic trees, where the Green's function can be expressed as an infinite sum of the Green's function over all possible trips between two points on a branching structure. By generalising Brownian motion measure on the infinite cable to finite, graphical spaces, they were able to derive a Feynmann path integral for diffusion along dendritic trees. In the limit of a large number of independent random walks in one dimension, Abbott *et al.*'s [1991] Green's function is Gaussian with mean $x - y$ and standard deviation $\sqrt{t}$:

$$G_0(x-y,t) = \frac{1}{\sqrt{4\pi t}}\, e^{-\frac{(x-y)^2}{4t}}\, \mathrm{e}^{-\frac{t}{\tau}}. \tag{3.108}$$

Note that this equation differs from (3.98) due to having a fixed diffusion constant $D = 1$ on all branches. Abbott *et al.* [1991] take D to be constant because they solve a version of the cable equation (3.1) where the spatial coordinate has been rescaled by a constant characteristic length λ, thereby fixing this parameter on all branches. In Section 3.5.1, we will show how to relax this assumption, to allow λ_i to be different for each branch i, and therefore allowing a different diffusion constant D_i for different i. Abbott *et al.* [1991] also consider the leakage term $\mathrm{e}^{-t/\tau}$ separately, not including it in their definition of $G_0(x-y,t)$. Because it has no spatial dependence, this has no effect on the derivation of the path integral. However, for continuity and simplicity, we include it here rather than reinserting it later.

For the semi-infinite cable where $x \geq 0$, with a closed tip boundary condition at $x = 0$ as in (3.60), Abbott *et al.* [1991] construct the Green's function for measurement at x and injection at y as a combination of two G_0 terms, effectively categorising the trips into those which touch the point $x = 0$, and those which do not. The latter class of trips can be computed simply as $G_0(x-y,t)$, never touching the origin. Any random walker that does touch $x = 0$, however, must reflect back towards positive x with probability 1, and thus must be treated differently. If we take the final part of the random walker's trip, from its final contact with $x = 0$ to the point y, and reflect this section about $x = 0$, we see that there exists a one-to-one correspondance between all trips from x to y that touch $x = 0$, and trips from x to the imaginary point $-y$. The Green's function for this type of trip is simply $G_0(x+y,t)$, and we can trivially construct the Green's function for all paths on a semi-infinite cable, from x to y in time t, by

$$G_{\text{closed}}(x,y,t) = G_0(x-y,t) + G_0(x+y,t). \tag{3.109}$$

We can construct the Green's function for a semi-infinite cable with an open boundary (3.59) at $x = 0$ in a similar manner. With this terminal condition, any path that touches $x = 0$ is lost and must be discounted. The Green's function for this system is therefore

$$G_{\text{open}}(x, y, t) = G_0(x - y, t) - G_0(x + y, t). \tag{3.110}$$

A first step towards generalising from the single cable to the tree is made by considering a single branching node, with n semi-infinite cables radiating from it, and a coordinate choice of $x = 0$ at the node for all segments. The random walkers behave normally along the cables, performing a random walk along them, and at $x = 0$, jump onto segment k with a probability p_k, given by

$$p_k = \frac{a_k^{3/2}}{\sum\limits_{i=1}^{n} a_i^{3/2}}, \tag{3.111}$$

where a_k is the radius of branch k. A more general definition for the probabilities p_k is given by

$$p_k = \frac{\left(\lambda_k\, r_k\right)^{-1}}{\sum\limits_{i=1}^{n} \left(\lambda_i\, r_i\right)^{-1}}, \tag{3.112}$$

allowing λ_i to vary according to the branch; this simplifies to Abbott *et al.*'s [1991] probabilities (3.111) if $\lambda_i = \lambda \, \forall \, i$, as is assumed in their work. Indeed, the 3/2-power in (3.111) stems from the dendritic input conductivity, as shown by Rall [1959], and can be recovered by taking R_{m} and R_{l} to be the same for all branches in (3.112), thereby imposing the same λ everywhere on the tree. If space is normalised by this fixed space constant, such that a generalised definition of the electrotonic distance x/λ is considered, the system of equations and boundary conditions describing the voltage about a node can be collapsed down onto the Green's function for the infinite cable, should the radii obey

$$\sum_{j \neq i} a_j^{3/2} = a_i^{3/2}, \tag{3.113}$$

where j are all daughter branches of i. Rall [1962a,b] used this relation in his derivation of the equivalent cylinder framework, a model used primarily in the simulation of cat motoneurons. This equivalent cylinder rule allows any tree satisfying the radial requirement to therefore be simplified to an analogous single cable, for which we can solve easily.

Let us now consider a node with multiple semi-infinite branches extruding from it, and the paths from a point x on branch i to a point y on branch j. When $i = j$, there is a contribution of $G_0(x - y, t) - G_0(x + y, t)$ from trips that do not touch the node, as with G_{open} in (3.110). Then, regardless of whether $i = j$, there is a contribution from those that do touch the node of $2p_j\, G_0(x + y, t)$, where the factor of 2 arises because of the analogous paths on the infinite cable, going from x to y and from x to $-y$, both with probability p_j. From this, we note that passing through a node from a branch i onto a branch j takes a factor of $2p_j$, while trips reflecting off a node back onto branch j pick up a factor of $2p_j - 1$ as coefficient. By continuing along these lines, Abbott *et al.* [1991] were able to demonstrate that their random walks, following the rules mentioned here, satisfied the required boundary conditions at all branching nodes and terminals. They were able to move away from random walks along the branches, to graphical *trips* between nodes. The result is that a Green's function $G_{ij}(x, y, t)$, for walks starting at x on branch i and finishing at y on branch j in time t, is constructed by an infinite sum over functions of all possible trips from x to y via any number of nodes.

3.5.1 Rules for Trip Construction

In their first paper on the dendritic path integral, Abbott *et al.* [1991] provide a set of rules for constructing valid trips on a dendritic structure. These are required to compute the Green's function on the structure, and hence to compute the voltage response to a stimulus on a particular dendritic tree. In his subsequent paper, Abbott [1992] provides a diagrammatic version of the rules, easily applicable to any given branching structure, and the final paper in the series provides a computational algorithm for the dendritic path integral [Cao and Abbott, 1993].

Trips are described as a sequence of node identifiers, sandwiched between the points x and y, from which trips begin and terminate, respectively. The trip $x\,A\,B\,y$ is valid only if the point x is adjacent to the node A, which must be adjacent to node B, which in turn must be adjacent to the point y, where adjacency is defined by the presence of a branch between the nodes. Trips have a length L_{trip}, the total distance travelled along the branches : that between x and A, added to the length of the branch (A, B), added to the distance between node B and point y. Finally, trips also have a coefficient A_{trip}, a product of the terms $2p_j$ and $2p_i - 1$, as introduced in Section 3.5. Initially, $A_{\text{trip}} = 1$. After the trip leaves x, if it passes through a node and onto branch j from a different branch i (such that $j \neq i$), it would take a factor of $2p_j$, while any trip reflecting back onto the same branch i at the node would take a factor of $2p_i - 1$. Therefore, each time the trip encounters a node coming from a branch i, its A_{trip} is multiplied by either $2p_j$ or $2p_i - 1$, depending on whether it had passed through and onto branch j, or reflected from the node back onto i. When the trip reflects off a terminal, A_{trip} is multiplied by $+1$ for closed terminals, or by -1 for open terminals. This is shown diagrammatically in Figure 3.11.

Trips are used to construct the Green's function $G_{ij}(x, y, t)$, for injection of current at a point y on branch j, and measurement at x on branch i. It is constructed by an infinite sum over functions of trips starting from x and finishing at y, of which there are an infinite number. Trips may leave x in either direction along branch i. They are allowed to change direction only at nodes, where they can pass onto another branch, or reflect back onto the same branch. At terminals, trips always reflect. Trips may visit nodes and may pass the point y any number of times, but must eventually end exactly at y on branch j. The Green's function is then defined as

$$G_{ij}(x, y, t) = \sum_{\text{trips}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}, t), \tag{3.114}$$

where $G_\infty(L_{\text{trip}}, t)$, defined in a manner similar to (3.98), but with its $(x-y)$ distance dependence replaced with a space-constant-normalised trip length L_{trip} :

$$G_\infty(L_{\text{trip}}, t) = \frac{1}{\sqrt{4\pi\, D_j\, t}}\, \mathrm{e}^{-\frac{(L_{\text{trip}})^2\,\tau}{4t}}\, \mathrm{e}^{-\frac{t}{\tau}}, \tag{3.115}$$

where $L_{\text{trip}} = L_{\text{trip}}(x/\sqrt{\lambda_i},\, y/\sqrt{\lambda_j})$ is the sum of the lengths of the edges travelled during the trip, normalised by their own space constants.

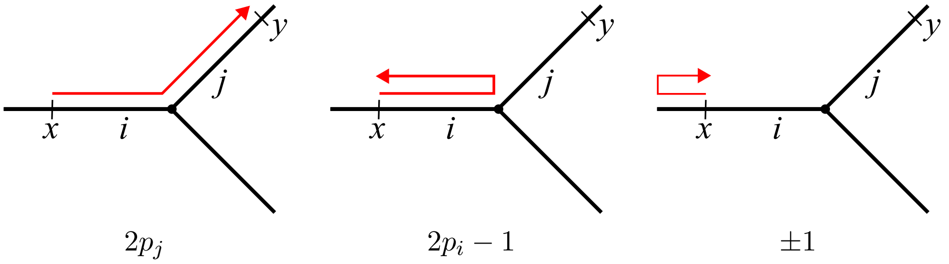


Figure 3.11: Factors taken to A_{trip} as a trip encounters a node. The quantities p are defined as in (3.111).

These rules are sufficient for the construction of the Green's function (3.114). The path integral framework also provides an identity for computing the Green's function for the exchange of points x and y :

$$G_{ji}(y, x, t) = \left(\frac{D_j\, r_j}{D_i\, r_i}\right)^{3/2} G_{ij}(x, y, t). \tag{3.116}$$

Due to the superexponential suppression of the $e^{-L_{\text{trip}}^2\tau/4t}$ term in (3.98) for large path lengths, Abbott suggests that the Green's function (3.114) can be truncated. This approximation is especially valid at short times. This implies that the short-time behaviour of passive voltage propagation on dendritic trees is dictated by the shortest paths between x and y, while distal parts of the tree have no impact at these times. For longer timescales, this sum-over-trips methodology samples from the wider tree and more trips are required for the path integral to converge as the exponential suppression becomes less significant. In addition, the number of possible trips grows exponentially with trip length. Abbott provides a brief analysis of the path integral's convergence [Abbott, 1992], stating that, for a trip visiting N nodes, the trip's length scales with N and hence, G_∞ scales with $e^{-\alpha N^2}$, with constant α. Assuming an infinite binary tree where each branch has equal radius (and hence $p_k = 1/3\ \forall\ k$), the Green's function G_{ij} is a sum of trips, each multiplied by their own A_{trip}. Assuming that a trip reflects off n nodes and passes through $N-n$ nodes, then

$$A_{\text{trip}}^{N,n} = \left(\frac{2}{3}\right)^{N-n}\left(-\frac{1}{3}\right)^{n}. \tag{3.117}$$

There are $\binom{N}{n}$ ways of selecting trips of length N with n reflections, and at each non-reflection, there are 2 ways the trip can go. The total sum of path coefficients A_{trips} is the sum over all possible combinations of reflections and transmissions through nodes, each weighted by the relevant coefficient, that is,

$$\sum A_{\text{trip}} = \sum_{n=0}^{N} 2^{N-n}\binom{N}{n}A_{\text{trip}}^{N,n}. \tag{3.118}$$

Putting these together, we find that

$$\sum A_{\text{trip}} = \sum_{n=0}^{N} \frac{2^{N-n}N!}{n!(N-n)!}\left(\frac{2}{3}\right)^{N-n}\left(-\frac{1}{3}\right)^{n} = \left(\frac{4}{3}-\frac{1}{3}\right)^{N} = 1, \tag{3.119}$$

and hence, the sum of coefficients A_{trip} does not grow with N, implying that the exponential suppression factor $e^{-L_{\text{trip}}^2/4Dt}$ does not have to compete with a growing number of trips; with longer trips exponentially more numerous but suppressed with e^{-L^2}, trips should contribute monotonically less to the Green's function (3.114) as length increases.

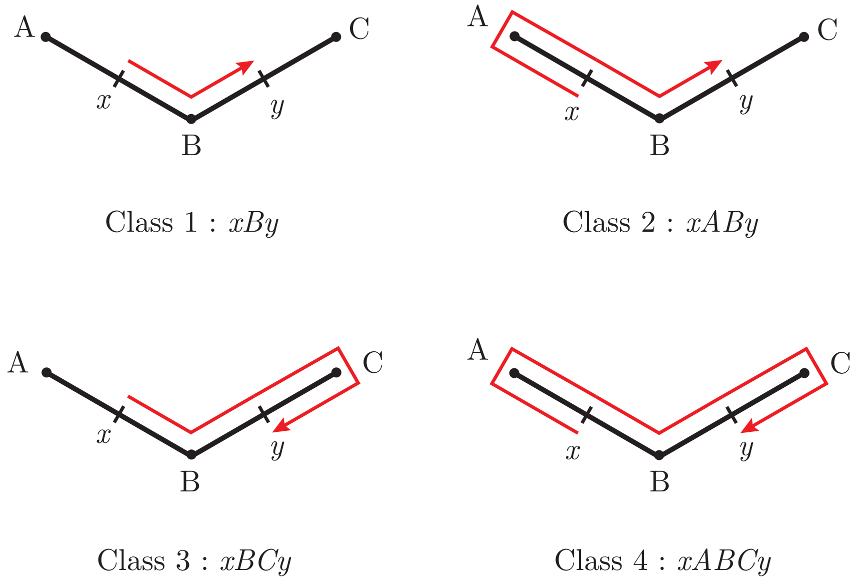
3.5.2 Trip Classes

Trips can be separated into four classes, with each belonging to one class. This serves to indicate the direction from which the trip leaves x and approaches y. On the infinite cable, there exists only the direct $x \to y$ trip, with all others going to infinity and not being able to change direction. This direct trip, leaving x in the direction of y and approaching y from the x direction, is a Class 1 trip, and has the trip description $x\,y$. On a branching structure, such as that in Figure 3.12, x and y can exist on the same branch (in which case $x\,y$ would still describe the most direct trip), or on different branches : the structure in Figure 3.12 has $x\,B\,y$ as its shortest, most direct Class 1 trip. Should a trip leave x and move *away* from y at first, before reflecting at a node or terminal and eventually arriving y as it moves away from x, the trip is said to belong to Class 2. On the figure, $x\,A\,B\,y$ is the shortest Class 2 trip. Class 3 trips move from x towards y at first, moving past the point, and finish by reflecting and approaching y by moving towards the point x. The shortest Class 3 trip on Figure 3.12 is $x\,B\,C\,y$. Any trip that leaves x by moving away from y, and finishes approaching y by moving towards x is a Class 4 trip. These are summarised visually in Figure 3.12. Trips are referred to as being *analogous* when they are constructed from the same core path along the tree, only allowing the first or last step to change, and hence, defining which class the trips belong to. It is possible to construct the shortest Class 2, 3 and 4 trips from the shortest and most direct trip between x to y; this is always a Class 1 trip. These newly-constructed paths are analogous with the shortest trip. Likewise, from any non-direct Class 1 trip, which performs excursions by visiting other nodes (potentially many times), it is possible to construct three analogous trips.

All analogous Class 2 and 3 trips have longer trip lengths L_{trip} than their analogous Class 1 trip. All Class 4 trips have longer L_{trip} than the analogous Class 1, 2 and 3 trips. However, depending on where along the branches the points x and y are placed, it is not possible to say, in general, which of the Class 2 or 3 trips are longer. Once determined for a particular branching structure, however, the relationship always holds : if, on a given tree, Class 2 trips are longer than Class 3 trips, this will always be true for that tree.

3.5.3 Theoretical Convergence and Term Ordering

We can show that Abbott's sum-over-trips makes an error that is bounded from above by e^{-N}, if all terms up to and including those of length N nodes are considered. We consider an identical diffusion coefficient D for all branches, although it is



Class 1 : xBy

Class 2 : $xABy$

Class 3 : $xBCy$

Class 4 : $xABCy$

Figure 3.12: The four classes of trips, as defined by Cao and Abbott [1993]. Class 1 is the most direct trip, leaving x in the direction of y, and not going past y before finishing (xBy). Class 2 leaves x in the other direction, but finishes when it meets y ($xABy$). Class 3 moves from x towards y, but goes past y and changes direction immediately after passing it before finishing ($xBCy$). Class 4 trips move from x first away from point y, and pass y before reflecting on the next node and finishing ($xABCy$).

possible to generalise this proof to support different diffusion coefficients. Fixing t throughout, we let

$$\begin{aligned} G_{ij}(x,y) &= \sum_{\text{trips}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}) \\ &= \sum_{k=0}^{\infty} \sum_{\substack{\text{trips with} \\ k \text{ nodes}}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}), \end{aligned} \tag{3.120}$$

where $G_\infty(L_{\text{trip}})$ is defined as in (3.98) for constant t.

We do not assume that all branches have equal radius. This implies that A_{trip} is a product of N factors $2p_i \in (0,2)$ and $2p_i - 1 \in (-1,1)$, where N is the number of nodes visited by the trip. Then,

$$|A_{\text{trip}}| \leq 2^N. \tag{3.121}$$

There exists a constant $B > 0$ such that every trip touching N nodes satisfies

$$L_{\text{trip}} \geq BN. \tag{3.122}$$

This makes B the coefficient of the lower bound on trip length, in terms of the number of nodes in a trip. Intuitively, B is the minimum distance between any two nodes, where, for this purpose, we count x and y as nodes.

Let F_N be the number of trips with N nodes. For any realistic, finite, dendritic morphology, each node has degree $d \leq 3$, and the total number of trips with N nodes is bounded by

$$F_N \leq 3^N. \tag{3.123}$$

We wish to classify trips by the number of nodes they meet. We therefore introduce Γ_N, the Green's function for trips with a given number of nodes, defined by

$$\Gamma_N = \sum_{\substack{\text{trips with} \\ N \text{ nodes}}} A_{\text{trip}}\, G_\infty(L_{\text{trip}})\ . \tag{3.124}$$

Then

$$\begin{aligned} |\Gamma_N| &= \left| \sum_{\substack{\text{trips with} \\ N \text{ nodes}}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}) \right| \\ &\leq \sum_{\substack{\text{trips with} \\ N \text{ nodes}}} \Big| A_{\text{trip}}\, G_\infty(L_{\text{trip}}) \Big| \\ &= \sum_{\substack{\text{trips with} \\ N \text{ nodes}}} \big| A_{\text{trip}} \big| \, \big| G_\infty(L_{\text{trip}}) \big|. \end{aligned} \tag{3.125}$$

For simplicity, we rewrite the Green's function (3.98) as $G_\infty(L_{\text{trip}}) = C\mathrm{e}^{-EL_{\text{trip}}^2}$, where

$$C = \frac{\mathrm{e}^{-t/\tau}}{\sqrt{4\pi Dt}}, \qquad \text{and} \qquad E = \frac{1}{4Dt}\ . \tag{3.126}$$

Then using (3.121) - (3.125), we obtain that the absolute value of the node-dependent

Green's function Γ_N is bounded from above :

$$\begin{aligned} |\Gamma_N| &\leq \sum_{\substack{\text{trips with} \\ N \text{ nodes}}} 2^N C \mathrm{e}^{-EL_{\text{trip}}^2} \\ &\leq \sum_{\substack{\text{trips with} \\ N \text{ nodes}}} 2^N C \mathrm{e}^{-EB^2N^2} \\ &\leq F_N \, 2^N C \mathrm{e}^{-EB^2N^2} \\ &\leq 3^N 2^N C \mathrm{e}^{-EB^2N^2} \\ &= 6^N C \mathrm{e}^{-EB^2N^2} \\ &= C \mathrm{e}^{-N(EB^2N - \ln(6))}. \end{aligned} \tag{3.127}$$

We define an integer $M = \lfloor \ln(6)/(EB^2) \rfloor$, such that the term in the exponential in (3.127), $EB^2N - \ln(6)$, is positive for all path lengths N greater than M. Then

$$\begin{aligned} \left| \sum_{N=0}^{\infty} \Gamma_N \right| &\leq \sum_{N=0}^{\infty} |\Gamma_N| \\ &\leq \sum_{N=0}^{\infty} C \mathrm{e}^{-N(EB^2N - \ln(6))} \\ &= \sum_{N=0}^{M} C \mathrm{e}^{-N(EB^2Nk - \ln(6))} + \sum_{N=M+1}^{\infty} C \mathrm{e}^{-N(EB^2N - \ln(6))}. \end{aligned} \tag{3.128}$$

The first sum in (3.128) is a finite sum of finite terms, and is hence finite. We will now show that the second sum is also finite using d'Alembert's ratio criterion for convergent series. The ratio ρ_N of the consecutive terms in the series, N and $N+1$, is

$$\begin{aligned} \rho_N &= \left| \frac{C \mathrm{e}^{-(N+1)(EB^2(N+1) - \ln(6))}}{C \mathrm{e}^{-N(EB^2N - \ln(6))}} \right| \\ &= \mathrm{e}^{-(EB^2(2N+1) - \ln(6))} \end{aligned} \tag{3.129}$$

Letting $N \to \infty$, we obtain

$$\begin{aligned} \rho_\infty &= \lim_{N\to\infty} \rho_N \\ &= \lim_{N\to\infty} \mathrm{e}^{-(EB^2(2N+1)-\ln(6))} \\ &= 0. \end{aligned} \tag{3.130}$$

With $\rho_\infty < 1$, the second sum in (3.128) converges absolutely for all constants $B, C, E > 0$. Therefore, the series in (3.128) is absolutely convergent for sufficiently-high N.

If we define $G_{ij}^M(x,y,t) = \sum_{N=0}^{M} \sum_{\substack{\text{trips with}\\ N \text{ nodes}}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}, t)$, then

$$\left|G_{ij} - G_{ij}^M\right| \le \sum_{N=M+1}^{\infty} C\mathrm{e}^{-N(EB^2N-\ln(6))} \tag{3.131}$$

and the path integral converges faster than e^{-N} in the worst case, with the number of nodes N visited by the trips.

This analysis attempts to put bounds on the magnitude on how A_{trip} scales with N, but the only major assumption made is that there exists some non-zero distance between any two nodes, that is, that no two nodes exist on top of each other. In a well-constructed dendritic tree, this assumption is perfectly valid, as nodes represent points taken from a three-dimensional reconstructed image of a dendritic tree. This assumption allows us to quantify, in terms of a bound, the minimum length of a trip that meets N nodes. The parameter which defines at which point we truncate the series solution, and how much error we make in doing so, is therefore N. In contrast, Abbott's convergence analysis [Abbott, 1992] uses L_{trip} as a cut-off parameter, and in doing so, assumes that all trips are ordered by their lengths in a monotonically-increasing fashion, where the $n+1^{\text{th}}$ term in the series solution (3.114) therefore is smaller in absolute magnitude than the n^{th} term. We will define the trip length at which the sum-over-trips is truncated as L_{trunc} : any trip with $L_{\text{trip}} > L_{\text{trunc}}$ will not be included in the series solution.

Because the number of trips grows exponentially with trip length, whether characterised by L_{trunc} or N, the order in which the series is constructed becomes incredibly important for the convergence of the solution. With the trip coefficients A_{trip} being impossible to compute without explicitly constructing each trip (except on very simple structures), a heuristic must be used to determine which subset of the infinite number of trips need to be generated. With convergence demonstrated as a function of increasing length, then computing the Green's function can be done by generating the trips in order of increasing length, and hoping that, with this ordering, their impact on the sum (3.114) does not deviate too strongly from the optimally-ordered sum, where each term contributes less than its predecessor.

In order to construct the terms in order of their length, Cao and Abbott [1993] derived an algorithm based on finding the shortest path between x and y (a Class 1 path on any acyclic structure), and computing the analogous Class 2, 3 and 4 paths from that. They then add an *excursion* to the Class 1 path, or a deviation from the shortest path, and compute the other analogous classes of trip from this one. While, for large L_{trunc}, this approach yields a generally monotonic series of trips in terms of their lengths, it does not guarantee a strict monotonicity, nor does it generally yield an increasing series over small ranges of L_{trunc}. To illustrate this point, we consider the simple branching structure in Figure 3.13A.

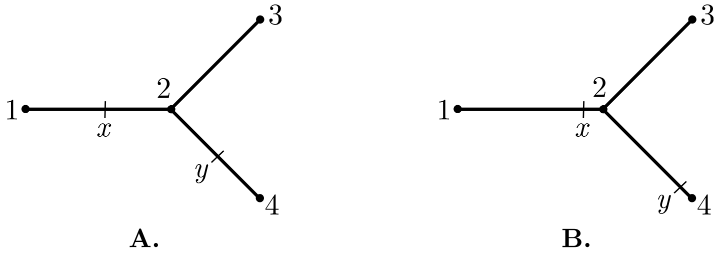


Figure 3.13: A model dendritic structure. **A** has the points of injection y and measurement x near the middle of their respective branches, whereas **B** has x shifted towards y, and y shifted away from x, which will cause trips of Class 2 to be longer than those of Class 3.

The shortest trip from x to y along this structure is described by the sequence of nodes, $x\,2\,y$. This Class 1 trip will generate three new trips, one for each of the remaining classes. The associated Class 2 trip is $x\,1\,2\,y$, the Class 3 trip is described by $x\,2\,4\,y$, and the Class 4 trip is $x\,1\,2\,4\,y$. If we associate the same length $L = 1$ with each branch, and we place points x and y in the middle of their respective branches, then we have generated four trips with the lengths $L^1_{\text{trip}} = 1$, $L^2_{\text{trip}} = 2$, $L^3_{\text{trip}} = 2$, $L^4_{\text{trip}} = 3$. Should we add an excursion to the Class 1 trip, say, $x\,2\,3\,2\,y$, and generate the analogous Class 2, 3 and 4 trips ($x\,1\,2\,3\,2\,y$, $x\,2\,3\,2\,4\,y$ and $x\,1\,2\,3\,2\,4\,y$ respectively), then we obtain lengths that continue to monotonically increase.

The situation changes, however, should we place x and y at different locations. If we consider the tree in 3.13B, we note that Class 2 trips will repeatedly be longer than Class 3 trips. We move x such that it is closer to node 2 than to node 1, and we move y in the same direction, making it closer to 4 than to 2, then the Class 2 trip will be longer than the Class 3 trip. Indeed, the shortest Class 2 trip $x\,1\,2\,y$ has length $L_{\text{trip}} = 2.8$ while the shortest Class 3 trip, $x\,2\,4\,y$ has length $L_{\text{trip}} = 1.2$, assuming we place x and y a distance 0.1 from their closest nodes. This loss in monotonicity will occur along any four analogous, consecutive trips : it is a product of generating trips in class order as opposed to true length order. The loss in trip length monotonicity is therefore due to an assymmetry between where x and y are located along their edges. Generating trips in terms of their classes will lead to a period-4 oscillation in trip lengths, which would not occur if trips were generated in ascending order of their lengths, as shown in Figure 3.14.

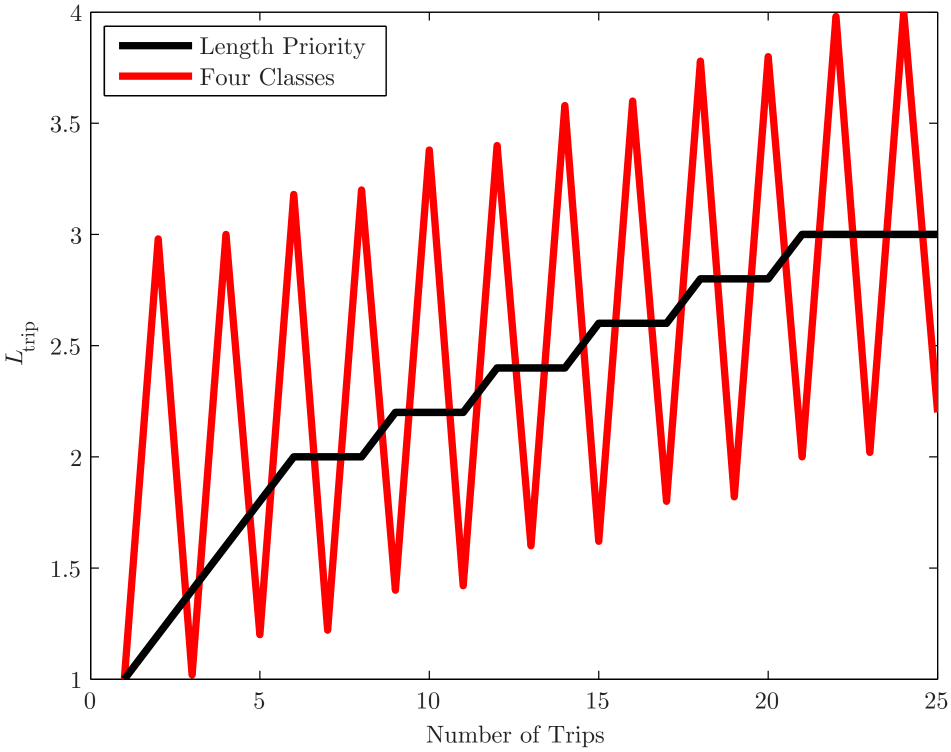


Figure 3.14: Trip lengths, as ordered by the algorithm proposed in Cao and Abbott [1993], in red. The length of trips, sorted ascending by their lengths, is shown in black.

Pathological structures can be found wherever an assymmetry exists in the tree. The oscillatory generation of trips by the length, artificially-induced by the four classes paradigm, is one example of a pathology in the tree. As we will see in the next chapter, other assymmetries may cause different problems in computational convergence, depending on the order in which the trips are constructed. In theory, however, the path integral approach is convergent, and can therefore be used to compute the Green's function solutions for certain cable structures, in closed form.

3.6 Closed-Form Solutions for Simplified Structures

With the sum-over-trips demonstrated to converge on any dendritic morphology for a sufficiently-high number of terms, at least in theory, we can begin to consider the analytical solutions for certain simplified geometries, where the Green's function can be written explicitly. In the following expressions, we are able to calculate A_{trip} analytically for the given geometry, which allows us to bypass the issue of having to generate trips according to the length heuristic described above. As in Abbott *et al.*'s [1991] derivation of the path integral, we will assume that $\lambda_i = \lambda$ for all branches i, and hence, that there exists only one diffusion constant D over the entire tree, for symmetry.

3.6.1 Finite Single Cable

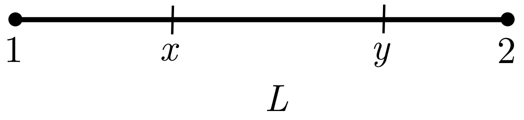


Figure 3.15: Finite cable of length L.

Consider a finite cable of length L, with points of measurement and injection x and y respectively, as in Figure 3.15. If we assume closed boundaries at $x = 0$ and $x = L$, then the Green's function can be written trivially as

$$\begin{aligned} G(x,y,t) = \sum_{n=0}^{\infty} \Big[& G_\infty\big(2nL + y - x, t\big) \\ & + G_\infty\big(2nL + y + x, t\big) \\ & + G_\infty\big(2(n+1)L - y - x, t\big) \\ & + G_\infty\big(2(n+1)L - y + x, t\big) \Big], \end{aligned} \tag{3.132}$$

where $G_\infty(x,y,t) \equiv G_\infty(x-y,t)$ is defined in (3.98). This equation contains four terms in the sum, conveniently separating the trips into their relevant classes. This is possible because the structure allows us to explicitly compute the length of the trips as a function of the possible reflections on the tree. There are four possible configurations of trips, which correspond to the four classes of trip. Beyond these, longer trips are simply excursions from the direct $x \to y$ path, for example, $x\,2\,1\,y$ (the second-shortest Class 1 trip, corresponding to the shortest trip with a single excursion). There exists only one length, L, and each excursion implies a reflection on both terminals, for a total additional length of $2L$. We can then write the length of the trips, L_{trip}, as a function of the number of excursions the trip makes, n. For this simple structure, A_{trip} remains 1 for all trips, assuming closed boundaries at the terminals.

3.6.2 Star Graph Cells

A star graph contains one central node, which we will label 0, and B external nodes, each connected only to the central node. If $B = 1$, the structure is simply the finite cable, although our definition of star graph will impose that $B \geq 2$; we consider $B = 1$ as a separate case, treated in Section 3.6.1. Two examples of star graphs with $B = 2$ and $B = 6$ can be seen in Figure 3.16. Here, we derive a novel combinatorial counting scheme for the construction of a closed-form solution to the dendritic path integral for symmetrical star graphs.

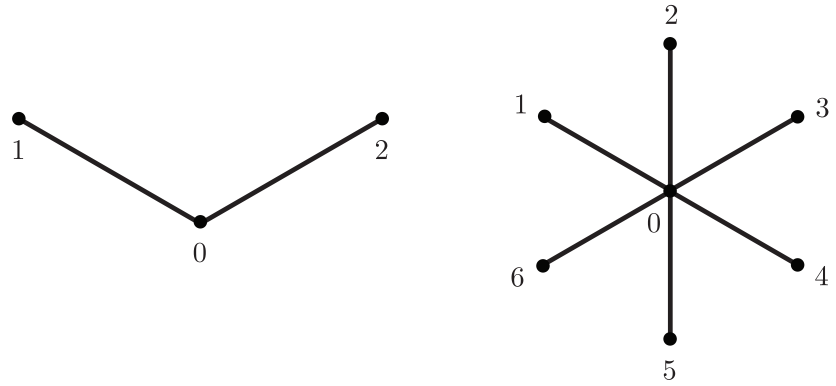


Figure 3.16: $B = 2$ and $B = 6$ star graphs, with x and y.

We assume that each branch has the same length L, and that the radius of each branch is equal, that terminals are closed and that x and y reside on different branches. Path lengths can be computed as in the case of the finite single cable : if the points x and y are measured by their distance from the central node 0, then the trips lengths are simply

$$\begin{aligned} L^1_{\text{trip}}(n) &= \quad x + y + 2nL, \\ L^2_{\text{trip}}(n) &= -x + y + 2(n+1)L, \\ L^3_{\text{trip}}(n) &= \quad x - y + 2(n+1)L, \\ L^4_{\text{trip}}(n) &= -x - y + 2(n+2)L. \end{aligned} \tag{3.133}$$

Unlike for the finite cable, trip lengths are not sufficient to compute the Green's function for star graphs. Because trips can now pass through and reflect off non-terminal nodes, we must compute the trip coefficients A_{trip}. With all branches having equal radii, we have $p = 1/B$, as defined in (3.111), independent of the branch being moved onto, and therefore, $2p = 2/B$ is the coefficient taken when we pass through the central node onto another branch, and $2p - 1 = (2 - B)/B$ is the coefficient that arises from reflecting off the central node. As before, due to our closed boundaries at the terminals, the coefficient taken from reflecting off a terminal is 1.

We begin by considering, for a trip consisting of n excursions, how many trips can be made where all excursions are reflections. With x and y on different branches, there is at least one factor of $2p$, to get us from x on a given branch to y on another. If we have one excursion ($n = 1$), then we can reflect at the beginning ($x\,0\,1\,0\,y$), or at the end ($x\,0\,2\,0\,y$).

In order to generalise to any n, we introduce the concept of a *transition* : if we consider a sequence of nodes visited during a trip, a transition occurs when we move onto another branch. They therefore exist as a triplet of nodes; any triplet where the first and last nodes are not equal is a transition. Any move that is not a transition must necessarily be a reflection, having the first and last nodes equal to one another. For these purposes, we must substitute x and y with whichever node is not adjacent to them in the triplet, in order to maintain the direction of trip movement. An example is the trip $x\,0\,1\,0\,2\,0\,y$, a Class 1 trip with $n = 2$ excursions (as $x\,0\,y$ is the shortest trip). Here, the nodes perform a reflection ($x \to 0 \to 1$, substituted with $1 \to 0 \to 1$), then a transition ($1 \to 0 \to 2$), then another reflection ($2 \to 0 \to y$, substituted with $2 \to 0 \to 2$).

If we are interested in computing the number of trips which can contain only one transition, for a sequence of nodes with n excursions, we note that this transition must be the one that brings us from the branch where x resides, to the one which hosts the point y. There are, therefore, $n+1$ possible places to put a transition, and with only one transition, then there are $\binom{n+1}{1}$ arrangements of nodes. The number of trips with the coefficient $A_{\text{trip}} = (2p)(2p-1)^n$, therefore, is

$$q\Big((2p)(2p-1)^n\Big) = \binom{n+1}{1}, \tag{3.134}$$

where $q(A_{\text{trip}})$ denotes the number of trips with that coefficient.

We can now consider trips with n excursions that experience only two transitions, that is, those with coefficient $A_{\text{trip}} = (2p)^2(2p-1)^{n-1}$. By the same logic, n excursions and two transitions can be arranged in $\binom{n+1}{2}$ possible ways. However, we must now note that only one transition is *definite* : the first transition has to leave the branch which hosts x but may now move onto any branch except the branch hosting y, while the second transition may move from any branch except the branch hosting x (as it has moved away from this branch), and must move onto the branch containing y. The first transition is indefinite, having the possibility to move onto any of $B-2$ branches (all except the branch hosting x and the branch hosting y), while the second transition is definite, because it must move onto the branch containing y. The *multiplicity* of the $\binom{n+1}{2}$ locations to arrange the transitions is therefore $B-2$, and hence, the number of trips with coefficient $A_{\text{trip}} = (2p)^2(2p-1)^{n-1}$ is

$$q\Big((2p)^2(2p-1)^{n-1}\Big) = \binom{n+1}{2}(B-2). \tag{3.135}$$

The number of trips with three transitions, or $A_{\text{trip}} = (2p)^3(2p-1)^{n-2}$, begin to show more complicated expressions. There are $\binom{n+1}{3}$ to arrange their transitions along the length of the trip, but their multiplicity is no longer so trivial. We first consider the case where the first transition moves the trip onto the branch containing y. The second transition then has $B-1$ branches to choose from (all except the branch it is currently on, including the branch hosting x). This transition hence takes us away from the branch hosting y. The third transition is the definite transition, taking us onto the branch containing y. This scenario has multiplicity $B-1$. The other situation is when the first transition moves onto any branch except the one containing y. As it is a transition and thus cannot reflect back onto the same branch, this leaves $B-2$ choices of branch. This implies that the second transition also has $B-2$ choices : any except the branch it is currently on, or the branch containing y, as this branch is reserved for our definite transition. The total multiplicity is this $(B-1)(B-2)^2$, and the total number of trips with n excursions, of which three are transitions, is

$$q\Big((2p)^3(2p-1)^{n-2}\Big) = \binom{n+1}{3}(B-1)(B-2)^2. \tag{3.136}$$

It is easier to continue enumerating these possibilities by using a decision tree. Figure 3.17 shows the decision tree for calculating the multiplicity of trips with n excursions, of which five are transitions (such that $A_{\text{trip}} = (2p)^5(2p-1)^{n-4}$). The decisions for each transition occur left-to-right; moving upwards implies a choice where the trip moves onto the branch containing y (and hence, all upwards decisions have multiplicity 1), while moving downwards implies that the trip will not move onto the branch hosting y during this transition, which may have multiplicity $B-1$ or $B-2$, depending on whether we are allowed to move onto any non-y branch, or any branch that must also be different to the previous branch.

By following the possible paths in these decision trees, we are able to construct solutions for trips with four and five transitions. We find that

$$q\Big((2p)^4(2p-1)^{n-3}\Big) = \binom{n+1}{4}\Big(2(B-1)(B-2)+(B-2)^3+(B-2)^2\Big), \tag{3.137}$$

and

$$q\Big((2p)^5(2p-1)^{n-4}\Big) = \binom{n+1}{5}\Big((B-1)^2+3(B-1)(B-2)^2+(B-2)^4\Big), \tag{3.138}$$

where the latter result can be checked from the decision tree in Figure 3.17 by multiplying the multiplicity of each decision as we move down the tree, and summing over all final branches.

The number of trips for any coefficient A_{trip}, can thus be found as a product of two terms. The first is a combinatorial expression in the number of excursions n the trip makes and the number of transitions, k, made during the trip. Here, $1 \leq k \leq n+1$, as the direct trip (with zero excursions, $n = 0$) has a single transition. The second is a polynomial in the number of branches in the star graph, with coefficients dictated by the number of transitions k made during the trip. By expanding the polynomials, we can observe a pattern :

$$\begin{array}{lccccccccccl}
k=1 & & & & & 1 & & & & & & n \geq 0 \\
k=2 & & & & 1B & & +2 & & & & & n \geq 1 \\
k=3 & & & 1B^2 & & -3B & & +3 & & & & n \geq 2 \\
k=4 & & 1B^3 & & -4B^2 & & +6B & & -4 & & & n \geq 3 \\
k=5 & 1B^4 & & -5B^3 & & +10B^2 & & -10B & & +5 & & n \geq 4
\end{array}$$

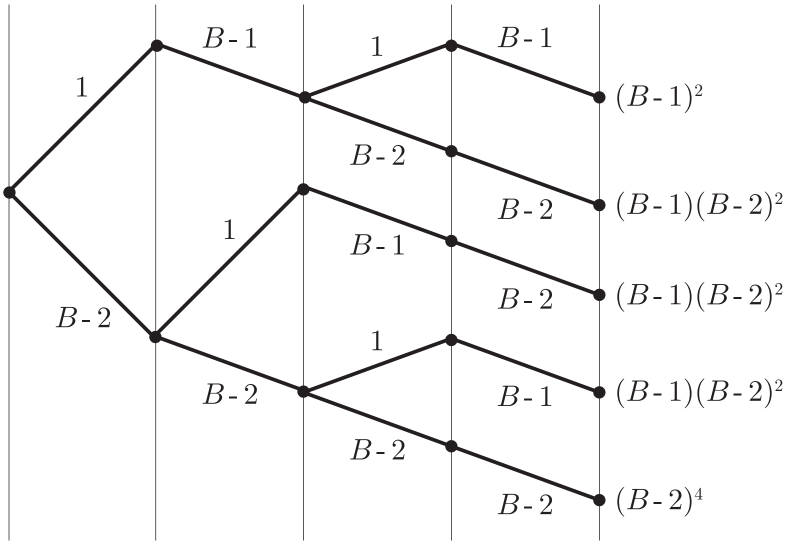


Figure 3.17: The decision tree for a trip with five transitions. Starting at the leftmost node, we decide whether the next transition will take us onto the branch containing y (an upwards choice), or any other branch (a downwards choice). We describe the middle trajectory as an example. Our first transition takes us away from the branch containing x, and onto a branch not containing y, allowing a total of $B-2$ choices. The next transition is onto the branch containing y, with multiplicity 1. Then, we have no choice but to move away from this branch at the next transition, onto any non-y branch, with $B-1$ choices. From here, our fourth transition must take us to a place where we can move onto the branch containing y on the fifth transition, and therefore must be any non-y branch that is different to our current branch. Hence, we have $B-2$ choices at this stage. The total multiplicity for trips following this pattern is then $(B-1)(B-2)^2$. An example Class 1 trip for this branch of the decision tree might be $x\,0\,3\,0\,3\,0\,2\,0\,4\,0\,1\,0\,1\,0\,y$, where we have five transitions and two reflections (a total of $n = 6$ excursions).

The coefficients to the polynomial expressions follow Pascal's Triangle, with the rightmost diagonal removed. If this is correct, then the polynomial can be expressed as

$$P(k) = \sum_{i=1}^{k} \binom{k}{i-1} (-1)^{i-1} B^{k-i}. \tag{3.139}$$

This combinatorial expression can be shown to generate the correct number of total trips, should we sum over all possible combinations of transitions and reflections. If we fix the number of excursions n, we have B^n total possible valid trips on a star graph with B branches. We then consider the number of trips with k transitions, and sum over k :

$$\sum_{k=1}^{n+1} q\Big((2p)^k(2p-1)^{n-k+1}\Big) = \sum_{k=1}^{n+1}\binom{n+1}{k}P(k). \tag{3.140}$$

Substituting $P(k)$ from (3.139), we find

$$\begin{aligned}
\sum_{k=1}^{n+1} q\Big((2p)^k(2p-1)^{n-k+1}\Big) &= \sum_{k=1}^{n+1}\binom{n+1}{k}\sum_{i=1}^{k}\binom{k}{i-1}(-1)^{i-1}B^{k-i} \\
&= -\sum_{k=1}^{n+1}\binom{n+1}{k}B^k\sum_{i=1}^{k}\binom{k}{i-1}\left(-\frac{1}{B}\right)^{i} \\
&= -\sum_{k=1}^{n+1}\binom{n+1}{k}B^k\sum_{j=0}^{k-1}\binom{k}{j}\left(-\frac{1}{B}\right)^{j+1} \\
&= \sum_{k=1}^{n+1}\binom{n+1}{k}B^{k-1}\sum_{j=0}^{k-1}\binom{k}{j}\left(-\frac{1}{B}\right)^{j} \\
&= \sum_{k=1}^{n+1}\binom{n+1}{k}B^{k-1}\left(\sum_{j=0}^{k}\binom{k}{j}\left(-\frac{1}{B}\right)^{j} - \left(-\frac{1}{B}\right)^{k}\right) \\
&= \sum_{k=1}^{n+1}\binom{n+1}{k}B^{k-1}\left(\left(1-\frac{1}{B}\right)^{k} - \left(-\frac{1}{B}\right)^{k}\right) \\
&= \frac{1}{B}\sum_{k=1}^{n+1}\binom{n+1}{k}\Big((B-1)^k-(-1)^k\Big) \\
&= \frac{1}{B}\sum_{k=0}^{n+1}\binom{n+1}{k}\Big((B-1)^k-(-1)^k\Big) \\
&= \frac{B^{n+1}-0}{B} \\
&= B^n\ ,
\end{aligned} \tag{3.141}$$

and thus, this expression finds the correct number of paths for any number of excursions n, for any number of transitions k and for any number of branches B on a star graph.

We can now construct an expression for the Green's function. We must consider all trips from $n = 0$ to N excursions (an upper limit on trip length), each of which must have from $k = 1$ to $n + 1$ transitions. Each of these has a given coefficient, A_{trip}, and we are now able to enumerate the number of trips with this coefficient, $q\big(A_{\text{trip}}\big)$. So, the Green's function for star graphs with B branches can be written

$$\begin{aligned}
G_{ij}(x,y,t) &= \sum_{\text{trips}} A_{\text{trip}}\, G_{\infty}(L_{\text{trip}},t) \\
&= \sum_{n=0}^{N} \sum_{\substack{\text{trips with} \\ n \text{ excursions}}} A_{\text{trip}}\, G_{\infty}(L_{\text{trip}},t) \\
&= \sum_{n=0}^{N} \underbrace{\sum_{k=1}^{n+1} q\big(A_{\text{trip}}\big)}_{\substack{\text{trips with} \\ n \text{ excursions}}} A_{\text{trip}}\, G_{\infty}(L_{\text{trip}},t) \\
&= \sum_{n=0}^{N} \sum_{k=1}^{n+1} \underbrace{\sum_{i=1}^{k} \binom{n+1}{k}\binom{k}{i-1}(-1)^{i-1}\, B^{k-i}}_{q\left(A_{\text{trip}}\right)} \underbrace{(2p)^{k}\,(2p-1)^{n-k+1}}_{A_{\text{trip}}} \\
&\quad \times \Big(G_{\infty}(L^{1}_{\text{trip}}(n),t) + G_{\infty}(L^{2}_{\text{trip}}(n),t) \\
&\qquad + G_{\infty}(L^{3}_{\text{trip}}(n),t) + G_{\infty}(L^{4}_{\text{trip}}(n),t) \Big),
\end{aligned} \tag{3.142}$$

which accounts for all classes of trips up to N excursions, with all possible combinations of reflections and transmissions, with the correct coefficients, and for the right trip lengths.

3.7 Conclusions

In this chapter, we presented a complete derivation of the cable equation, along with its assumptions and some of its steady-state solutions. We then derived the Green's function general solution by using Laplace and Fourier transforms, and used this form to introduce Abbott *et al.*'s [1991] dendritic path integral. We demonstrated how the path integral can be used to construct simple and extremely-efficient closed-form solutions for some simple dendritic structures, such as the finite cable and symmetrical star-graphs, for which we derived a novel and optimal solution.

Closed-form solutions such as those derived in Sections 3.6.1 and 3.6.2 provide a highly-efficient manner of computing the Green's function for certain branching structures. Equation (3.132) groups trips on the finite cable by their class, and creates each trip by increasing its base length by twice the length of the cell, an augmentation on the mother trip dubbed an excursion. For star cells, the solution in Section 3.6.2 is optimal, in the language of a sum-over-trips : equation (3.142) is a compact form of the dendritic path integral, in which trips are grouped by their coefficients, by the number of reflections and transitions they make. As such, trips are merely enumerated and not explicitly created, a phenomenal computational saving.

However, the morphological constraints imposed by these solutions do not lend these solutions to the study of realistic dendritic systems. Few neuronal systems exhibit the symmetry that these solutions rely on. Star graphs can be used to caricature systems such as the starburst amacrine cells in the retina, as in Figure 2.15. Linear structures, such as in Figure 3.15, have been used to simulate the flow of calcium between an array of cells [Harris and Timofeeva, 2010].

For more realistic dendritic morphologies, with complex branching structures, and not limited to a set of symmetries to be exploited in simplifying the solution, more general methods for construction of the Green's function must be considered. The next chapter introduces several algorithms for solving the dendritic path integral on arbitrary geometries, directly in the time domain – a minority amongst a plethora of Laplace-domain methods. These more general algorithms construct the Green's function by sampling from the full dendritic morphology, again, based on the work of Abbott *et al.* [1991] and the algorithmic implementation of Cao and Abbott [1993].

Chapter 4

Time-Domain Methods

The solutions introduced in the previous chapter are extremely valuable : those that are simple enough (and do not rely on combinatorial expansions) provide mathematical insight into the behaviour of the transmembrane voltage in cables; those more complicated still provide a very rapid means of computing the path integral and hence, to efficiently simulate voltage spread along branching structures. Unfortunately, the majority of branching structures are too complex to enable a compact, closed-form sum-over-trips solution to be derived, relying instead on the sampling of paths from the tree in a fashion we must decide upon.

The intuition given to us by Abbott [1992], which we supplemented with a more formal and general derivation in Section 3.5.3, states that the impact of a trip on the sum-over-trips solution diminishes exponentially with the number of trips, should they be ordered by increasing trip length; hence, we will first consider algorithms which sample trips from the tree in order of increasing trip length. We describe the classed-based excursions heuristic of Cao and Abbott [1993] in Section 4.2, and develop a novel formal language framework allowing for improvements to be made to this algorithm in Section 4.3.

From this point onwards, we introduce a set of new algorithms for sampling trips from dendritic trees. A true length-priority heuristic is developed in Section 4.4. Then, in Section 4.5, we introduce the Feynman-Kac formula for the cable equation on trees, and derive a Monte Carlo algorithm for it, sampling trips stochastically using a random walker on the nodes of the tree. In Section 4.6, we map the tree to a discretised analogue and compute the sum-over-trips using a modified incidence matrix to count trips with a given length and calculate trip coefficients on-the-fly. Section 4.7 compares the computational convergence of the algorithms detailed in this chapter, and Section 4.8 summarises these results.

4.1 Graph Theory and Algorithms Terminology

When dealing with dendritic structures, it is extremely useful to represent the branching tree as a graph, or network. The field of graph theory can provide a great deal of results useful to the following work, and can make concrete some of the concepts used. We therefore begin by a short list of graph-theoretic terminology.

Graph

A graph $\mathbf{G}$ is typically defined by a set of *nodes* or *vertices*, $\mathcal{V}$, and a set of *edges*, $\mathcal{E}$. Edges may be *directed* or *undirected*, and represent a relationship between a given pair of nodes. In addition to direction, edges may be given a *weight*, which may represent a cost or distance associated with moving from one node to its neighbour. An edge between vertices $i, j \in \mathcal{V}$ may be denoted $(i, j) \in \mathcal{E}$.

A graph may be fully defined by its *adjacency matrix* A, a square matrix denoting the distances between connected nodes. In the case of unweighted graphs, the matrix is binary, with $A_{ij} = 1$ if an edge exists between vertex i and vertex j, and $A_{ij} = 0$ otherwise. Undirected graphs have a symmetrical adjacency matrix, as an edge between i and j implies an edge between j and i, whereas, in directed graphs, $(i, j) \not\Longrightarrow (j, i)$.

A variant on the adjacency matrix is the *edge-adjacency matrix*, denoted B, where $B_{mn} = 1$ implies that edges $m, n \in \mathcal{E}$ now share a node.

Degree

The degree of a node $i \in \mathcal{V}$, denoted $d(i)$, is the number of edges adjacent to i. If the graph is directed, a node's degree can be classified according to whether the edges are incoming (*in-degree*) or outgoing (*out-degree*). A node i with $d(i) = 1$ is typically called a *terminal*; the *root node* is simply a designated special case of a terminal node. If a directed edge (i, j) points from a node i to a node j, then i is said to be the edge's tail, and j is the edge's head.

Tree

A tree is an acyclic graph. This also prohibits self-cycles, that is, $(i, i) \notin \mathcal{E}$ for any tree. A *binary tree* is a tree in which the maximal degree for any node is three. Equivalently, nodes in a binary tree have one *parent*, and no more than two *children.*

Walks

A walk is a sequence of nodes on the graph, where each node is connected by an edge to the node that precedes it. *Direct walks* follow the shortest available path between their first and last nodes; on a tree, there is only one direct walk between any two nodes. Any walk that is not direct is allowed to visit nodes that are not along the direct walk (an *excursion*), or visit nodes more than once. The term *path* will be used synonymously with *walk* in this work. In the dendritic path integral literature [Abbott *et al.*, 1991; Cao and Abbott, 1993; Coombes and Bressloff, 2003; Timofeeva, 2003; Svensson, 2009; Harris and Timofeeva, 2010], paths are primarily called *trips*.

Connectedness

A fully-connected graph is one in which there exists at least one walk between any two pairs of nodes. Graphs that are not fully connected have a number of connected *components* that is greater than one, each of which are fully-connected subgraphs.

Cardinality

Perhaps not truly terminology belonging to graph theory or algorithms, the cardinality of a set remains a central concept in these fields. Taking the set of edges $\mathcal{V}$ as an example, its cardinality is denoted $|\mathcal{V}|$, and measures the number of elements in the set – thus, $|\mathcal{V}|$ is the number of nodes on a tree. For any fully-connected tree, $|\mathcal{V}| = |\mathcal{E}| + 1$.

Spanning Tree

For any fully-connected graph defined by its set of nodes $\mathcal{V}$ and set of edges $\mathcal{E}$, a spanning tree is a graph consisting of all nodes in $\mathcal{V}$ and a subset of the edges in $\mathcal{E}$ such that all nodes lie on the tree and there exists a path between any two nodes, but no cycles are present. Any tree is its own spanning tree; any cyclic graph can admit multiple spanning trees, each of which form a tree which spans all nodes in $\mathcal{V}$. Any bridging edge (an edge that, if deleted, increases the number of connected components in the graph) must belong to the spanning tree.

Shortest Path Tree

A shortest path tree T for a graph $\mathbf{G}$ is a spanning tree with a root at a given node $v \in \mathcal{V}$, such that the length of the path between any node $u \in \mathcal{V}$ and v on the shortest path tree T is equal to the length of the shortest path between these nodes in $\mathbf{G}$. Shortest path trees are not necessarily unique.

Order

The total order of a tree is the total number of levels that make up the tree, counted from the root. The root is defined as having order zero, its children having order one, and so on, until the furthest terminal node.

Heap

A tree in which the nodes have some sort of value, or *key*, associated with them satisfies the min-heap property if, for all nodes, the parent node's key is less than or equal to the keys of any child nodes. Similarly, a max-heap is a tree in which the key of the parent node is greater than or equal to the keys of children nodes. Heaps are examples of ordered data structures, and are especially important in algorithmic graph theory as they implement the maximally-efficient form of a priority queue, a structure which sorts elements in a queue and always provides the element with the highest priority first.

Time Complexity

Mathematically, a graph is a representation of a set of objects, explicitly linked according to some relation between them. When viewed from a computational perspective, however, graphs can be considered data structures from which we would like to extract information using an algorithm : a procedure, or set of instructions, which computes a desired quantity or finds a function of interest in finite time. Some algorithms are faster at producing their output than others – however, the absolute speed with which an algorithm provides a solution to the required problem is dependent on the size of the dataset, on certain parameters of the dataset, on computational power available, on memory access-times, and many more factors which make defining an absolute compute time for an algorithm a meaningless endeavour. Instead, a theoretical speed, or *time complexity*, can be associated with an algorithm, as a function of the size of its input. If the algorithm acts on a graph, then its time complexity can be defined as a function of the number of nodes in the graph, for example, describing how the algorithm's performance *scales* with the size of its input.

Big $\mathcal{O}$ notation is used to describe the time complexity of an algorithm. An algorithm such as the Depth-First Search, which searches through a graph from a given node in depth order will, in the worst case, have to iterate through every node in order to find what it is searching for. This algorithm is therefore linear in the number of nodes on the graph, and has a time complexity denoted by $\mathcal{O}(|\mathcal{V}|)$. This states that the algorithm will slow linearly as we increase the size of the graph that must be searched. The Breadth-First Search algorithm, which searches through the graph by visiting all neighbours of a node before moving on in its search list, also has a time complexity of $\mathcal{O}(|\mathcal{V}|)$. Both of these algorithms can be used to find the shortest path between any two given nodes : they begin at the starting point, and continue searching until they have found the goal node.

Arguably the most famous shortest-path algorithm is Dijkstra's [1959] algorithm. It has a time complexity of $\mathcal{O}(|\mathcal{V}|^2)$ for sparse graphs, which means that it scales quadratically with the number of nodes on the graph. This compares disfavourably with Depth- and Breadth-First Search algorithms; however, it must be noted that this description of performance only captures the *scaling* of the algorithm. If we wish to know how the algorithm would perform in practice, we must consider two things : that an algorithm may typically perform far better than its worst-case running time, and that, to obtain actual running time, we must also consider the *constant* that precedes the performance scaling. Because Big $\mathcal{O}$ notation describes the algorithm's scaling behaviour, if an algorithm requires $10n^2$ time, and another requires n^2 time, both are described as being $\mathcal{O}(n^2)$ (where n is the size of the input), although the latter algorithm will always scale better than the prior.

Big $\mathcal{O}$ notation describes the theoretical worst scaling of an algorithm, providing an upper bound on the algorithm's scaling with input size n. For example, the time taken to find the smallest element in an array of size n is $\mathcal{O}(n)$. However, certain algorithms can typically operate much more rapidly than their worst-case complexity – in this example, it would take $n/2$ operations on average.

Finally, algorithms may operate in different time complexities depending on the problem. Using a Fibonacci heap to implement a priority queue and applying it to sparse graphs, Dijkstra's algorithm only takes $\mathcal{O}(|\mathcal{V}|\log|\mathcal{V}|)$ time. The efficiency of Dijkstra's algorithm means that it will be at the heart of some of the algorithms presented in Chapter 4.

4.2 The Four Classes Algorithm

From (3.131), we know that truncating the sum-over-trips solution (3.114) at N excursions produces an error that scales with e^{-N} for a sum where the terms are ordered monotonically in their lengths. Thus, with each term in the sum-over-trips corresponding to a unique trip, the error is minimised for any k trips if we use the shortest possible k trips from x to y on the tree. We can find a correspondence between N and Abbott's L_{cutoff}, the length of a trip beyond which we truncate the sum. If we are able to construct trips from the single shortest trip, and then incrementally increase in length until L_{cutoff} is met, we have computed the k shortest trips that exist under the length threshold.

In this section, we will describe the algorithm proposed by Cao and Abbott [1993] which implements the dendritic path integral reviewed in Section 3.5. We will see how it constructs paths that are pseudo-monotonic in length, in that they are only guaranteed get monotonically longer every four terms, due to potential asymmetries in where points x and y are placed; see Figure 3.14 for a comparison between this periodic phenomenon and the true length ordering of trips. We will compare this algorithm with our own Length Priority algorithm, designed to avoid this periodic behaviour while being more optimal; we will also point out how a different type of asymmetry in the tree can cause this algorithm to show highly-discontinuous convergence.

4.2.1 Implementation by Cao and Abbott

Intuitively, an algorithm based on finding trips in increasing order of length will start with the shortest trip between the two points of interest. It is possible to find the shortest, most direct trip between any two points on a tree, using a search algorithm such as Dijkstra's algorithm, centering its search at x, or even a breadth-first or depth-first search. On a tree, the shortest trip is guaranteed to be unique, and should the tree be fully connected, its existence is also guaranteed. Dendritic morphologies fulfill both of these conditions : should there be multiple individual dendritic trees, they remain connected at the soma, and nowhere do dendrites form cycles. Cao and Abbott [1993] proposed an algorithm which begins by finding this unique shortest trip, and then derives longer trips by adding all possible excursions to this trip, creating an entire batch of longer trips. Iteratively, new trips are used to derive still longer trips by considering all possible excursions along each trip. Using Figure 3.12 as a reference, we can describe excursions more formally : if A and B are adjacent nodes in a tree, then an excursion could be added to the trip $x\,B\,y$ to generate the trip $x\,B\,A\,B\,y$, representing a reflection on node B towards A, reflecting at the terminal A back towards B, passing through this node and finally onto point y. This process can be iterated indefinitely, generating a trip with two more nodes each time. Finally, if all trips generated in one batch are above some length threshold, the algorithm terminates.

However, this simple algorithm suffers from certain inefficiencies. The Four Classes algorithm generates duplicate trips, which must then be removed by a binary search through the list of existing trips for every new trip generated [Cao and Abbott, 1993], which takes $\mathcal{O}(k \log k)$ time overall, for k trips constructed. There are two different mechanisms by which duplicate trips are generated, and both mechanisms can be eliminated by applying simple restrictions to the choice of excursions applicable to a trip. We continue to refer to Figure 3.12, where, as an example of the first mechanism, we note that it is possible to generate the trip $x\,B\,A\,B\,C\,B\,y$ in two different ways from the shortest Class 1 trip, $x\,B\,y$:

$$\begin{aligned} &x\,B\,y \xrightarrow[\text{Excursion}]{B \to BAB} x\,B\,A\,B\,y \xrightarrow[\text{Excursion}]{B \to BCB} x\,B\,A\,B\,C\,B\,y, \\ &x\,B\,y \xrightarrow[\text{Excursion}]{B \to BCB} x\,B\,C\,B\,y \xrightarrow[\text{Excursion}]{B \to BAB} x\,B\,A\,B\,C\,B\,y. \end{aligned} \tag{4.1}$$

Due to the fact that the excursion may be added at any step in the trip (at the first or second B), the same trip may be generated multiple times.

The second mechanism by which duplicate trips are produced is the addition of excursions along the same branch, starting from either end. In the structure in Figure 3.12 we have both $A \to ABA$ and $B \to BAB$. Hence, we can generate $x\,A\,B\,A\,B\,y$ in two different ways (brackets added for clarity) :

$$\begin{aligned} &x\,A\,B\,y \xrightarrow{\overset{\text{Excursion}}{A \to ABA}} x\,(A\,B\,A)\,B\,y, \\ &x\,A\,B\,y \xrightarrow{\overset{\text{Excursion}}{B \to BAB}} xA\,(B\,A\,B)\,y. \end{aligned} \tag{4.2}$$

Finding all possible excursions for each trip rapidly becomes expensive. For a trip passing through n nodes, then it costs $\mathcal{O}(nd)$ time to find all excursions, where d is the maximum degree on the tree. This must be repeated for every trip in each batch of trips, with n increasing with every new batch. Because of this scaling, it is not possible to determine the algorithm's time complexity in full. At the first stage, when we have found the shortest path and wish to add all possible excursions, then the number of nodes in the shortest path, n, scales with the diameter of the tree, which itself is bounded by the number of nodes in the tree, $|\mathcal{V}|$. At this point, we have one path, so it costs $\mathcal{O}(n)$ to obtain the first batch of longer trips, where d is omitted because we can assume that the maximum degree on the tree is bounded. This batch of n trips contain duplicates; we must therefore apply a binary search, at the cost of $\mathcal{O}(n \log n)$ to prune these out of the path list. We nonetheless retain

$O(n)$ trips, of length $n+2$ nodes, to which we must apply all possible excursions. This iteration therefore costs $\mathcal{O}(n + n\log n)$ operations to produce. The next batch is generated by adding $\mathcal{O}(n+2)$ excursions to $\mathcal{O}(n)$ trips and then binary searching for duplicates, costing $\mathcal{O}\big(n(n+2) + n\log n\big)$, such that finding two batches of trips has cost a total of $\mathcal{O}\big(n^2 + 2n + n\log n + (n+2)\log(n+2)\big)$ operations, and therefore scales with a leading order term of $\mathcal{O}(n^2)$.

The trend in the leading-order term, then, is that each iteration costs an additional power of n. While the first iteration scales with $\mathcal{O}(n)$, the second requires $\mathcal{O}(n^2)$, the third $\mathcal{O}(n^3)$, and so on. This cost is somewhat offset by the fact that trips are generated in an exponentially-increasing number per batch; this algorithm therefore scales linearly in the number of trips it generates. Practically, however, the algorithm contains inefficiencies which may be improved upon. Because trips are constructed only in batches, we may massively overshoot the number of trips we require : due to the termination condition, at least twice as many trips as are necessary to meet the length cutoff are generated. The removal of duplicate trips, due to its log-linear scaling with the number of trips per batch, also increases the cost per batch. Therefore, whilst it is not possible to write down an accurate scaling for this algorithm, we can say that, due to pruning for duplicate trips, the leading-order time complexity is $\mathcal{O}(n\log n)$, and that the algorithm requires at least double the computation time necessary to complete the task.

Applying this algorithm naively to the shortest trip will lead to the construction of only Class 1 trips. We know that trips always come in four classes, and that any direct trip is a Class 1 trip; we also know that there exist three analogous trips that can be written down immediately, without the need to construct them algorithmically. Cao and Abbott [1993] use this fact, applying the algorithm independently to each of the shortest Class 1, 2, 3 and 4 trips. If this process is applied to every node on every trip with n and $n+1$ nodes, then every trip with $n+2$ and $n+3$ nodes will be generated. Thus, from the four shortest $x \to y$ trips on the tree, it is possible to construct all trips up to some threshold number of nodes in length explicitly. If our first Class 1 trip has n nodes, then the Class 2 and 3 trips always have $n+1$ nodes, and thus, this condition is fulfilled. The lengths and coefficients of all trips can then be calculated from their full trip descriptions, allowing the Green's function given by equation (3.114) to be approximated up to some length threshold. However, the inefficiencies we have demonstrated here can motivate the development of improvements to this algorithm in order to improve its scaling from log-linear in the number of trips generated. We will therefore present two modifications which will be sufficient to prevent the construction of duplicate trips, without any trips being missed, leading to an algorithm that will scale linearly in the number of trips generated.

4.3 A Formal Grammar for Paths on Graphs

The motivation behind the development of a language-theoretic algorithm comes from its ability to formalise the ideas used in the Four Classes algorithm, and allow for modifications to be made in order to improve its performance. The idea of increasingly-longer trips on a branching structure is closely related to the formation of sentences in a linguistic sense, and the rules which dictate which trips on a tree are valid can be expressed in terms of a formal grammar. Before we describe the Four Classes algorithm in terms of formal language theory, and discuss modifications which will avoid the generation of duplicate trips, we must introduce some jargon.

4.3.1 Some Language Theory Terminology

Alphabets and Words

An alphabet Σ is a set of symbols which form the basis of any language. Alphabets can consist of a finite set of *letters*; these can be the alphabetical characters in the English language, or numbers, or any other set of symbols.

A word over an alphabet is any finite string, or sequence, of letters from the alphabet Σ. The set of all possible finite-length words, including the empty string, is denoted Σ^*. Words can be concatenated to form new words; concatenating a word with the empty word results in the original word.

Grammar

A grammar is a set of rules for the construction of words from letters in an alphabet. Based on a starting string or letter, the grammar provides *production rules* which, when applied, replaces a portion or the entirety of the word with another string. We have already introduced the syntax of production rules in Section 4.2.1, where excursions are described as, for example, $A \to ABA$. This implies that if the letter A is found in a word, it can be replaced by the series of letters, ABA.

A grammar that results in the construction of the same word using different production rules, as can be seen in (4.2), is an *ambiguous* grammar. The fact that the Four Classes algorithm provided in Cao and Abbott [1993] can be described by an ambiguous grammar is one of the reasons why it generates duplicate trips.

A *context-free* grammar is one where the left-hand side of all production rules consists of a single letter.

Language

A formal language over an alphabet Σ, is a subset of the words found in Σ^* which can be constructed using a given grammar. In terms of trips on a branching structure, where the alphabet $\Sigma = \mathcal{V}$ is the set of nodes on the tree, then the words in the language are the valid trips between x and y; the grammar relevant to obtaining the language is a set of rules which allow valid excursions to be applied to trips such that the resulting trips are valid with regard to the tree's morphology.

4.3.2 The Improved Four Classes Algorithm

As we noted, the major costs associated with Cao and Abbott's [1993] Four Classes algorithm are the generation of duplicate trips, which leads to a log-linear cost in trip generation, where it could potentially be linear; and the termination condition, which results in the algorithm generating at least twice as many trips as are required. By describing the algorithm in the theory of formal languages, we can address the prior issue at the core of the algorithm, as is described in the rest of this section. The latter issue can be eliminated entirely by generating trips one at a time as opposed to in exponentially-growing batches. Whilst this is only a constant factor, irrelevant to the algorithm's asymptotic scaling, it has practical implications for the algorithm's runtime as well as for its memory usage, which will later become a critical factor.

We can derive a formal language approach by requiring a grammar to construct words over the alphabet of the set of nodes, $\mathcal{V}$. Each word will constitute a valid trip on the tree. The production rules that make up the grammar are simply substitutions of the form $A \to ABA$, if and only if A is adjacent to B on the graph, and $A, B \in \mathcal{V}$. Because the tree is undirected, the existence of the rule $A \to ABA$ in the grammar implies the existence of $B \to BAB$, which, as we noted in (4.1), leads to the construction of the same trip in two different manners. This problem can be avoided by assigning each branch a direction. If the branch AB is given direction $\overrightarrow{BA}$, then the excursion $A \to A|BA$ is disallowed. The choice of direction for each branch is unambiguous on acyclic structures: apart from the branch on which x is found, each branch must be directed away from x. The branch upon which x resides is directed away from y. This ensures that each node has a sequence of excursions that allow the algorithm to generate trips including it. The allocation of direction to each branch can be performed before the process of generating trips, and may coincide with finding the four main classes of trips. These modifications require that the graph be acyclic, since “away from a point” is not generally definable on a graph with cycles. There do exist cyclic graphs for which an unambiguous grammar can generate the language of $x \to y$ trips, but these are not relevant to the study of single dendritic trees.

The other mechanism by which duplicate trips are created, shown in (4.2), can also be avoided. If we insist that excursions cannot be added at any step that precedes the excursion most recently added to the trip, this can be prevented. In the theory of context-free grammars, this is equivalent to requiring a leftmost derivation. We can represent this using the symbol | to separate the mutable and immutable parts of the trip :

$$x \,|\, B\, y \quad \xrightarrow{\substack{\text{Excursion}\\ B \to B|AB}} \quad x\, B \,|\, A\, B\, y \quad \xrightarrow{\substack{\text{Excursion}\\ B \to B|CB}} \quad x\, B\, A\, B \,|\, C\, B\, y, \tag{4.3}$$

$$x \,|\, B\, y \quad \xrightarrow{\substack{\text{Excursion}\\ B \to B|CB}} \quad x\, B \,|\, C\, B\, y \quad \not\xrightarrow{\qquad\quad} \quad x\, B\, A\, B\, C\, B\, y. \tag{4.4}$$

The two excursions in (4.3) will construct the path $x\,B\,A\,B\,C\,B\,y$, whereas the excursion sequence in (4.4) is unable to do so, because the second excursion could only be added to the B after the separation symbol, leading to the production of $x\,B\,C\,B\,A\,B\,y$ instead. Note that only the rightmost mutability symbol | is kept – anything to the left of this symbol cannot be used as a basis for an excursion.

These two modifications of the Four Classes algorithm are sufficient to prevent the generation of any duplicate trips, without any trips being missed. Together, they provide an unambiguous context-free grammar generating the entire language of $x \to y$ trips.

4.3.3 Application of the Improved Four Classes Algorithm

We will illustrate the application of the Four Classes algorithm, with the language-theory improvements mentioned above, on a simple branching structure. Consider the tree in Figure 4.1.

Initialisation

We must first attribute a direction to each edge on the tree. Following the convention described in Section 4.3.2, where each edge is directed away from x, and the edge containing x is directed away from y, we obtain an analogous directed graph whose

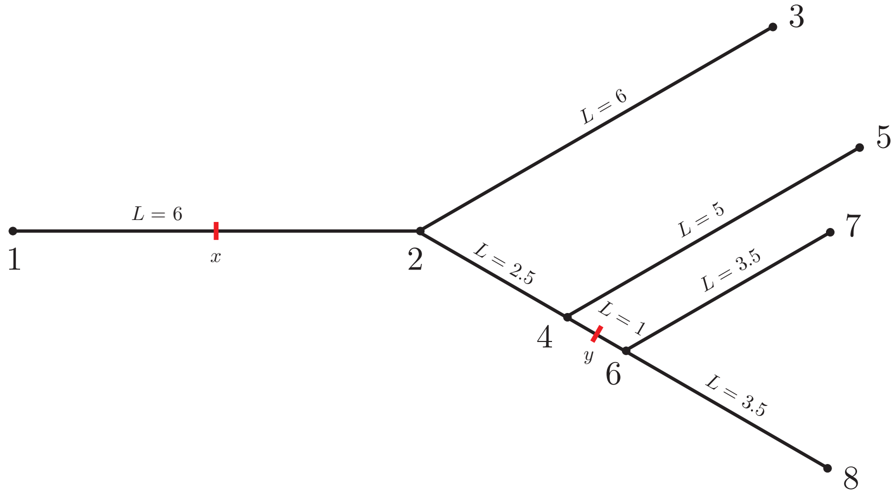


Figure 4.1: Example tree for the Four Classes algorithm.

edge set $\mathcal{E}$ is

$$\mathcal{E} = \Big\{(2,1),(2,3),(2,4),(4,5),(4,6),(6,7),(6,8)\Big\},$$

where an edge (A, B) is directed $A \to B$, with $A, B \in \mathcal{V}$. As before, the existence of $(A, B) \in \mathcal{E}$ implies that $(B, A) \notin \mathcal{E}$.

The set of directed edges $\mathcal{E}$ is sufficient to create the complete list of production rules. Each edge (A, B) defines a production rule, $A \to A|BA$. These must be ordered by their lengths such that we may easily select the shortest excursion at any time :

$$\begin{aligned}
4 &\to 4\,|\,6\,4, && L = 2,\\
2 &\to 2\,|\,4\,2, && L = 5,\\
6 &\to 6\,|\,7\,6, && L = 7,\\
6 &\to 6\,|\,8\,6, && L = 7,\\
4 &\to 4\,|\,5\,4, && L = 10,\\
2 &\to 2\,|\,3\,2, && L = 12,\\
2 &\to 2\,|\,1\,2, && L = 12.
\end{aligned}$$

The initialisation stage of the algorithm is concluded by finding the shortest trip from x to y, and associating with it a list of valid excursions. By using Dijkstra's algorithm, the shortest trip is found to be $x\,|\,2\,4\,y$. The trip therefore passes through two nodes, 2 and 4. Whilst production rules are only applied to Class 1 trips, such as this one, we can still immediately write down three analogous trips. From this Class 1 trip description, we therefore construct the analogous Class 2 trip by allowing $x\,2$ to become $x\,1\,2$, giving us the trip $x\,1\,2\,4\,y$. Similarly, we construct the analogous Class 3 trip by allowing $4\,y$ to become $4\,6\,y$, yielding $x\,2\,4\,6\,y$. By applying both transformations, we obtain the Class 4 trip, $x\,1\,2\,4\,6\,y$. We will keep these analogous trips in a separate list, as they will not be used to generate new trips, and as such, are not required to contain the | symbol for differentiating the mutable and immutable parts of the trip. Finally, we provide the Class 1 trip with a list of valid excursions. To be valid, an excursion is required to be based on a node present in the trip. We therefore associate with this trip a list of the excursions which have a left-hand side equal to 2 or 4. This *trip-excursions pair* is placed in a list, ordered by the length of the trip plus the length of its shortest excursion. With respect to the computational implementation of this algorithm, the trip-excursions pair (TEP) will be placed in an ordered container such that the first entry always corresponds to the TEP with the shortest combined trip length and shortest excursion. These data structures have an insertion cost of $\mathcal{O}(\log n)$ for n items in the list, ensuring that a new TEP is guaranteed to be inserted in the correct place along the list to ensure the list remains sorted. In this example, the shortest trip has a length of $L_{\text{trip}} = 6$, and its shortest excursion has $L = 2$, associating with it a combined TEP length of $L_{\text{TEP}} = 8$.

Continuation

The iterative part of this algorithm loops over the counter k, the number of trips generated. After the Initialisation, the counter is set to $k = 1$, and the Continuation stage will therefore be iterated until $k = k_{\text{max}}$, some maximum number of trips we wish to produce, or until the last trip generated exceeds a length of L_{cutoff}.

We begin by selecting the shortest item from the TEP list, an operation costing only $\mathcal{O}(1)$ time as the TEP list is a sorted container. This item represents the mother trip that will be used to generate a longer one, and its own list of valid excursions. The excursions list is already ordered, so we simply select the first one and apply it to the trip, yielding the guaranteed next shortest trip along the tree.

In our example, we had just entered the Continuation stage, and the TEP list only contained one item. We therefore select it, and apply the shortest excursion $(4 \to 4\,|\,6\,4)$ to this trip to construct the daughter trip $x\,2\,4\,|\,6\,4\,y$ as the second-shortest Class 1 trip. As before, we can construct analogous Class 2, 3, and 4 trips based on the Class 1 trip description by allowing transformations of the beginning and ending of the node sequence. We place these trips in our separate list, along with the previously-derived Class 2, 3, and 4 trips.

Once a new Class 1 trip has been generated by applying an excursion to a shorter trip, we must provide it with a list of valid excursions and place it in the TEP list. We note that our new Class 1 trip $x\,2\,4\,|\,6\,4\,y$ passes through nodes 2, 4, and 6, and therefore select all excursions whose left-hand side is equal to these. In this case, this corresponds to all possible excursions. We therefore associate the excursions list with the trip, and place it in the TEP list.

The final operation of the Continuation stage is the modification of the valid excursions list associated with the mother trip. In order to prevent the mother trip from being selected and generating the same daughter trip, we must remove the excursion used to generate this daughter trip. The first trip therefore now has a list of only four valid excursions :

$$\begin{gathered}\xcancel{4 \rightarrow 4\,|\,6\,4},\\ 2 \rightarrow 2\,|\,4\,2,\\ 4 \rightarrow 4\,|\,5\,4,\\ 2 \rightarrow 2\,|\,3\,2,\\ 2 \rightarrow 2\,|\,1\,2.\end{gathered}$$

The daughter trip is not affected by this removal, and contains its own list which, for this example, happens to contain the complete list of all excursions.

The insertion of the new trip-excursions pair into the TEP list may have modified the ordering of the list. Indeed, in our example, the TEP lists consists of two objects : the trip $x\,|\,2\,4\,y$ and its shortest valid excursion $2 \rightarrow 2\,|\,4\,2$; and the trip $x\,2\,4\,|\,6\,4\,y$ along with its shortest excursion, $4 \rightarrow 4\,|\,6\,4$. The first has an updated TEP length of $L_{\mathrm{TEP}} = 11$, while the daugther trip, of length $L_{\mathrm{trip}} = 8$ and with a shortest valid excursion of length $L = 2$, has a TEP length of $L_{\mathrm{TEP}} = 10$, placing it above the mother trip in the TEP list.

After incrementing our trip counter ($k \leftarrow k + 1$), this iteration of the Continuation stage is now finished. The next iteration would repeat this procedure : it would select the trip from the TEP list having the shortest TEP length (in our example, the trip $x\,2\,4\,|\,6\,4\,y$), would apply its shortest excursion $4 \rightarrow 4\,|\,6\,4$ to give $x\,2\,4\,6\,4\,|\,6\,4y$ with $L_{\mathrm{trip}} = 10$, would associate with it its own list of valid excursions and insert it into the TEP list with a value of $L_{\mathrm{TEP}} = 12$. The mother trip would see its shortest excursion made invalid, taking its L_{TEP} from 10 to 13. The TEP list now consists of three trips, the first of which is our original shortest trip, $x\,|\,2\,4\,y$

with its shortest valid excursion $2 \to 2\,|\,4\,2$, and thus having $L_{\text{TEP}} = 11$. Each trip is used to derive three analogous Class 2, 3, and 4 trips, placed in a separate list. Trips are constructed with constant cost and maintained in the TEP list with a cost of $\mathcal{O}(2 \log n)$, where one logarithmic factor comes from insertion of the daughter trip, and the other from the modification the excursions list of the mother trip to invalidate the excursion used this iteration.

Termination

The algorithm terminates when the next shortest trip to be constructed has a length that exceeds our predetermined cutoff length, L_{cutoff}. Because L_{TEP} is the length of the next shortest trip to be generated, we can halt the algorithm when the first item in the TEP list satisfies $L_{\text{TEP}} > L_{\text{cutoff}}$. Alternatively, we could terminate the algorithm after a fixed number of trips k_{max} were constructed. With either termination condition, we have constructed k trips, where k is the counter that is incremented during the Continuation stage. We copy each trip into the separate list of the Class 2, 3, and 4 trips which were constructed during the Continuation stage, and remove the | symbol from each (as these are no longer required). Then, because each algorithmically-constructed trip generated a total of four trips on the tree, we have constructed the shortest $4k$ trips between x and y as a cost of $\mathcal{O}(k \log k)$, with no duplicate trips constructed and no excess computation made.

Conclusions

The Improved Four Classes algorithm is a modification of Cao and Abbott's [1993] suggested implementation, which removes the generation of duplicate trips and a one-time inefficiency in the total number of trips generated for a requested threshold length. In doing so, the algorithm runs more efficiently, although the time complexity remains $\mathcal{O}(n \log n)$.

4.4 The Length Priority Algorithm

The length-priority heuristic for trip construction is motivated by the $\mathrm{e}^{-L_{\text{trip}}^2}$ term in the Green's function (3.98) and indicates that the path integral (3.114) may be made convergent if trips are generated in length order. An algorithm able to rapidly construct trips in a length-ascending order would therefore be highly beneficial in computing the Green's function on a tree. Our proposed improvements on the Four Classes algorithm have led to a method which avoids the duplicate construction of trips and reduces the computation time by a constant factor. However, despite being a more elegant algorithm for trip construction, by generating trips individually and uniquely, it remains log-linear for the construction of individual trips. In addition, the Four Classes algorithm and its improved derivative both require $\mathcal{O}(kn)$ space, for k trips with n nodes, which can impose a limit on the number of trips that are able to be constructed, given a specific hardware architecture.

Fortunately, the k shortest trips problem has interested graph theorists for many years, and as a result, many algorithms have been proposed [Dreyfus, 1969; Hoffman and Pavley, 1959; Yen, 1971]. A powerful algorithm by Eppstein [1999] provides both time and memory advantages, and can be used instead of the formal grammar derived in Section 4.3.2. The k shortest trips can be constructed on any branching structure, in $\mathcal{O}(|\mathcal{V}|+|\mathcal{E}|\log|\mathcal{E}|+k)$ time, assuming no edges have negative weights. It is therefore linear in the number of trips constructed, k, an improvement on our previous log-linear time.

Eppstein's algorithm circumvents an $\mathcal{O}(k^2|\mathcal{V}|)$ minimum for the listing of k trips explicitly by using an implicit trip description. By using a pointer to the previous trip and listing the details of the excursion, the algorithm stores trips as nodes in a tree, where each edge denotes an excursion added to a previous trip. Each node has at most a degree of $|\mathcal{V}|$, and the tree satisfies the heap-structured property such that, for all nodes in the tree, each child node represents a trip that is longer than its parent node. This implicit representation only requires $\mathcal{O}(k)$ space for k trips. A graph representing all possible excursions is then built. The shortest path tree and the excursions graph are constructed in $\mathcal{O}(|\mathcal{V}|+|\mathcal{E}|\log|\mathcal{E}|)$ time; from these, the k shortest paths can be found in linear time.

The algorithm also provides a method for computing any properties that can be described by a monoid in $\mathcal{O}(1)$ time per trip. Because coefficients are products of real numbers, the axioms required for an algebraic structure to be described as a monoid (associativity, closure and the existence of the identity element), then the coefficients A_{trip} may be described as monoids. Eppstein therefore offers the computation of trip coefficients in $\mathcal{O}(k)$ for k trips, an improvement on the Four Classes algorithm's $\mathcal{O}(kn)$ to compute the product for n nodes over k trips.

Jiménez-Marzal Optimisation

It was noted by Jiménez and Marzal [2003] that the construction of a graph of all possible excursions was computationally expensive. Thus, despite the Eppstein [1999] algorithm having the lowest worst-case time complexity of known k shortest path algorithms, the practical applicability of the algorithm was significantly reduced by this one-off initialisation cost. They went on to propose a recursive function which constructs only the parts of the excursions graph as they are required. While maintaining Eppstein's worst-case time complexity, this improvement reduces the amount of computation time considerably [Jiménez and Marzal, 2003] by turning it into a lazy algorithm.

Conclusions

By replacing the trip excursions concept in sampling from the tree with Eppstein's [1999] algorithm for finding the k shortest trips, we improve the asymptotic time complexity to one of $\mathcal{O}(k)$, linear in the number of trips constructed. An improvement in the amount of space required is also obtained, and the algorithm provides a method for computing trip coefficients in constant time per trip. A practical optimisation by Jiménez and Marzal [2003] allows even further speed improvements to be made, rendering this algorithm much faster than the Improved Four Classes method.

4.5 Monte Carlo Method

The path integral formulation of the solution to the cable equation introduced by Abbott *et al.* [1991] is derived via consideration of a Feynman-Kac representation of the solution in terms of random walkers on the dendritic geometry. Hence, it is natural to consider Monte Carlo approaches to evaluating this path integral. Instead of a length-ordered series solution as provided by the Length Priority approach, the Green's function (3.114) can be constructed using a stochastic algorithm. The aim of this approach is to sample from trips $x \to y$ in such a way that the probabilistically more likely samples coincide with the trips that contribute most to the series solution (3.114).

4.5.1 Random Walkers and Diffusion

To motivate this Monte Carlo approach, we consider a linear diffusion equation along an infinite one-dimensional cable,

$$\frac{\partial \mathcal{G}}{\partial t} = D\frac{\partial^2 \mathcal{G}}{\partial x^2}, \qquad -\infty < x < \infty, \;\; t \in [0,T], \tag{4.5}$$

satisfying the initial condition $\mathcal{G}(x,0) = \delta(x-y)$. Instead of solving this equation analytically or numerically, its solution can be found as the expectation of a stochastic process. Analogous to (4.5), a diffusion process for the state variable X_t can be defined by the stochastic equation

$$\mathrm{d}X_t = \sqrt{2D}\,\mathrm{d}W_t, \tag{4.6}$$

with the initial condition $X_0 = y$, and where W_t is a Wiener process.

Equation (4.5) is the Kolmogorov equation of the diffusion process (4.6); it can be described as the time evolution equation of the probability density for the state of the diffusion (4.6). This means that, should we solve (4.5) via classical numerical or analytical methods, we would discover the probability density of X_t. The relationship, fortunately, goes both ways : the expectation of the stochastic process (4.6) is described by the solution to (4.5), and thus, repeated sampling from the diffusion process (4.6) will converge onto the solution $\mathcal{G}(x,t)$.

Therefore, if we correctly set appropriate boundary conditions at branching nodes and terminals, this method of sampling from random walks can be applied to branching structures with arbitrary geometries, allowing us to compute $\mathcal{G}(x,t)$ on trees. We can then easily find a solution of the cable equation on this geometry using the relation

$$G_{ij}(x,y,t) = \mathcal{G}_{ij}(x,y,t)\;\mathrm{e}^{-t/\tau}.$$

The Feynman-Kac relation between the partial differential equation (4.5) and the Brownian motion (4.6) is at the heart of a Monte Carlo method which we will now describe in terms of discrete random walkers on branching structures.

4.5.2 Random Hoppers

We have established that the expectation of a function on random walks on the branching points of the dendritic tree is equivalent to the sum-over-trips form of the solution, (3.114). However, the sampling of random walkers diffusing along the edges of a tree is incredibly computationally expensive. In addition, with the constraint made by Abbott *et al.* [1991] that all branches have constant radius (with discrete jumps in radius at the nodes), the simulated continuous-space diffusion of the walkers would provide no benefit to the accuracy of the solution. Thus, the reduction of the random walk problem from the complete continuous space geometry of the neuron to the discrete topology of the branching nodes of the tree gives a considerable efficiency saving to a Monte Carlo solver. We therefore make a distinction between random walkers on the continuous space of dendritic branches, and random hoppers, which take discrete hops between the branching and terminal points of the tree.

We introduce a parameter $h_{\max}$, the maximum number of discrete hops on nodes for which we wish to calculate the expectation. The maximum number is based upon the effective maximum range of diffusion during the interval $[0, t]$, and is linked to the trip cutoff length L_{cutoff} introduced in Cao and Abbott [1993], as an upper limit on the length of trips. Then, we generate a realisation of a random walk on the nodes,

$$\omega = (\omega_1, \omega_2, \ldots, \omega_{h_{\max}}), \tag{4.7}$$

where each ω_h is a label identifying a particular node. For trips $x \to y$ we select ω_1 such that it is either of the two nodes adjacent to the branch containing x, with equal probability. By indexing a branch existing between two nodes, ω_{h-1} and ω_h, as the hth branch, subsequent steps are performed with the transition probability

$$P(\omega_h \mid \omega_{h-1}) = p_h, \qquad 2 \leq h \leq h_{\max}, \tag{4.8}$$

where p_h is given by (3.111). This connects the Monte Carlo method with the other algorithms derived from the dendritic path integral [Abbott *et al.*, 1991].

4.5.3 Obtaining the Green's Function Solution

The random hoppers allow us to sample from the tree's geometry, in the same way that a length-priority method does; here, however, we do not use length to guide our sampling, but rather the likelihood of the trips themselves. We then use properties of the trips, namely their lengths L_{trip} and their coefficients A_{trip} in a function (3.114) which establishes a link between the trips and the Green's function solution.

In the Monte Carlo method, each trip is constructed with a certain probability. In fact, because each random hopper takes a step onto a branch h with probability p_h, the probability of generating a particular trip in (4.7), is simply

$$P(\omega) = \prod_{h=2}^{h_{\max}} p_h, \tag{4.9}$$

or the product of the factors p_h. Thus, trips generated by the Monte Carlo method contain additional information, which may be harnessed by introducing two auxiliary functions, ϕ and $\tilde{a}$, of subwalks of ω. A subwalk is defined simply as any fraction of the walk (4.7) starting from its first node ω_1 and ending early at ω_h for some $h < h_{\max}$.

The first function serves to indicate whether a subwalk of h steps on a realisation ω is a valid trip, and is defined by

$$\phi(\omega, h, x, y, t) = \begin{cases} G_\infty\Big(L(\omega, h, x, y), t\Big), & \text{if } \omega_{h-1} \text{ and } \omega_h \text{ are adjacent to } y, \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

Here, $L(\omega, h, x, y)$ is the length of the trip starting at x, taking the trajectory set out by the subwalk of ω with h hops, and terminating at the point y. Therefore, if this subwalk of ω terminates by hopping across the edge which houses the point y, then ϕ returns the Green's function G_∞ for a trip of the correct length, and zero otherwise.

The other auxiliary function, $\tilde{a}$, is defined as

$$\tilde{a}(\omega, h) = \begin{cases} 1, & \text{if } h = 1,\ 2, \\ 2, & \text{if } \omega_{h-2} \neq \omega_h, \\ (2p_h - 1)/p_h, & \text{if } \omega_{h-2} = \omega_h, \\ 1, & \text{if at a closed terminal (taking priority).} \end{cases} \tag{4.11}$$

The relevant function on paths can be defined as a composite of the auxiliary functions described above :

$$\tilde{A}(\omega, x, y, t) = \sum_{h=1}^{h_{\max}} 2\ \phi(\omega, h, x, y, t) \left(\prod_{i=1}^{h} \tilde{a}(\omega, i) \right).$$

We can demonstrate that taking the expectation of $\tilde{A}$ with respect to the random walk (4.8) is equivalent to solving for the path integral, up to some value of $h_{\max}$ at time t :

$$\begin{aligned} \mathbb{E}_P\left[\tilde{A}(\omega, x, y, t)\right] &= \sum_{\omega} P(\omega)\ \tilde{A}(\omega, x, y, t) \\ &= \sum_{\omega} \sum_{h=1}^{h_{\max}} 2\ P(\omega)\ \phi(\omega, h, x, y, t) \left(\prod_{i=1}^{h} \tilde{a}(\omega, i) \right) \\ &= \sum_{\substack{\omega\,: \\ x \to y \\ \text{at } h}} \sum_{h=1}^{h_{\max}} 2\ P(\omega)\ G_\infty\Big(L(\omega, h, x, y), t\Big) \left(\prod_{i=1}^{h} \tilde{a}(\omega, i) \right) \\ &= \sum_{\substack{\text{trips} \\ x \to y}} A_{\text{trip}}\ G_\infty(L_{\text{trip}}, t), \end{aligned}$$

where $P(\omega)$ is the probability of the realisation ω, and $\mathbb{E}$ denotes the expectation operator. Therefore, the Monte Carlo strategy is to sample, sequentially or in parallel, the random function $\tilde{A}$ in order to construct this expectation.

Breaking the walk into subwalks offers the computational advantage that, for every trip realisation ω with $h_{\max}$ hops, we can truncate the walk at any $h < h_{\max}$ to obtain other walks.

Conclusions

A Monte Carlo on the space of valid excursions allows trips to be sampled probabilistically rather than by order of length. By randomly hopping between nodes on the tree, trips may be found in a highly parallelisable manner, yielding computational savings over high performance architectures.

4.6 Trip-Grouping Matrix Algorithm

In each of the algorithms thus far mentioned, trip are constructed and stored, either explicitly or implicitly, such that we obtain a complete description of the trip (a sequence of nodes), for each trip. For the Four Classes algorithm and its language-theoretic derivative, the vast majority of space required for the algorithm is accounted for by these trip descriptions. The memory consumed scales with $\mathcal{O}(kn)$, for k trips with n nodes. As k increases, and n increases with it, this rapidly become prohibitive. A trivial optimisation to these algorithms is to compute the important properties of the trips, L_{trip} and A_{trip} upon the construction of each trip, and simply store these, reducing the memory required to a more reasonable $\mathcal{O}(k)$, although these are still computed in $\mathcal{O}(kn)$ time. The Length Priority algorithm, based on Eppstein's [1999] algorithm, only requires constant space per trip due to its implicit trip representation, and is also able to compute both trip length and coefficient in constant time.

Nevertheless, the aforementioned methods are trip enumeration algorithms, constructing every one of the k trips required. This general class of methods will, at best, construct trips linearly in k, their core objective being the counting and construction of individual trips. In order to achieve sublinear times for trip construction, we must group the trips by some common characteristic.

An alternative method of constructing the sum-over-trips series solution, therefore, is by grouping trips by their lengths :

$$\sum_{\text{trips}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}, t) = \sum_{l} G_\infty(l, t) \sum_{\substack{\text{trips with} \\ L_{\text{trip}} = l}} A_{\text{trip}}, \tag{4.12}$$

where the sum over l is over all possible trip lengths L_{trip}. On a dendritic tree with branches described by a continuous space, trips grouped by their length makes little sense : groups may well only ever contain one trip unless the tree exhibits some symmetries, and as the group size tends to one, the sum (4.12) becomes equivalent to the individual trip construction of the path integral in (3.114). However, on a discretised dendritic tree, such as those in Rall's [1964] compartmental model or those discretised into *segments* by numerical packages such as NEURON [Carnevale and Hines, 2006], trip groups may become large. In this situation, the grouping of trips according to their lengths allows us to count the number of trips of a given length l without having to explicitly construct them individually.

The adjacency matrix is naturally suited to counting walks between two nodes on a graph. The elements nth power of the adjacency matrix A represent the total number of walks between two nodes. That is, $(A^n)_{ij}$ denotes the number of walks between i and j in n hops. The Trip-Grouping Matrix method uses a matrix Q similar to a directed edge adjacency matrix which encodes for coefficients.

4.6.1 Discretisation of the Dendritic Tree

In order to obtain a large number of trips grouped by their lengths, the branching structure is discretised into small, directed branches of length Δx. Then, maintaining our notation, $u, v \in \mathcal{V}$ are nodes that define the extremities of the discrete compartments of length Δx. Then, if u and v are adjacent, $e = (u, v) \in \mathcal{E}$ is a directed edge from u to v. For any edge $e = (u, v)$, we denote the reverse edge by $e' = (v, u)$. The existence of $e \in \mathcal{E}$ implies the existence of $e' \in \mathcal{E}$, that is, all edges are directed but all adjacent nodes are connected by two edges, one in each direction. In addition, the edge set $\mathcal{E}$ is a totally-ordered set of directed edges. Whilst the actual order of the edges in the set is not important, the existence of a binary operation on the set's elements which places them in some unambiguous order is crucial.

The points of measurement and injection x and y must reside along different edges; the discretisation length Δx must be chosen such that at least one node falls between x and y. This will simplify the computation of the Green's function once trips are counted.

Trips are taken to begin from a point x along a starting edge $s = (s_1, s_2)$, and end at a point y along a goal edge $g = (g_1, g_2)$ for $s, g \in \mathcal{E}$. For simplicity, we say that $x \in s$ or $x \in (s_1, s_2)$ if x resides along edge $s = (s_1, s_2)$. Because edges are directional, and two exist between any pair of adjacent nodes, the points x and y are no longer unique. In fact, if $x \in s$ resides along the starting edge s, then an entirely equivalent point $x \in s'$ exists along the reverse edge, and the same is true of the point y. The artificial directionality of edges in this method is introduced such that we may capture the direction of movement of the trips, again without explicitly constructing them : because the coefficient taken when a trip passes through a node is different to that taken when the trip reflects off a node, we must maintain directional information in the method. Because $x \in s$ and $x \in s'$ are equivalent, there is no need to differentiate between them.

The orientation of s and g are defined such that the shortest $x \to y$ trip satisfies $x \to s_2 \to \cdots \to g_1 \to y$. Therefore, the shortest $x \to y$ trip always starts on edge s, that is, in the $s_1 \to s_2$ direction, and approaches y along the edge g, in the $g_1 \to g_2$ direction. This is equivalent to a Class 1 trip; Class 2 trips leaving x along the $s' = (s_2, s_1)$ edge, arriving at $y \in g$; Class 3 trips go from $x \in s$ to $y \in g'$; Class 4 trips, finally, go from $x \in s'$ to $y \in g'$. The locations of the points $x \in s$ and $y \in g$ along their respective edges are given as a fraction of the branch length, such that $x\Delta x$ denotes the distance from x to s_2 and $y\Delta x$ is the distance between node g_1 and point y.

We must distinguish between k, the number of edges travelled in a particular trip, and the length of the trip L_{trip}. Because x and y reside along their respective edges, the total length of a trip that travels along k edges is less than if the full distance along k edges had been travelled. For example, a trip $x\, s_2\, A\, B\, C\, g_1\, y$ consists of $k = 6$ hops, but while the four central hops cover full edges, and thus have a length of Δx each, the first and last hops have a length of $x\Delta x$ and $y\Delta x$ respectively. Thus, $L_{\text{trip}} < k\Delta x$ for any combination of x, y and for all k.

4.6.2 Construction of the Edge-Adjacency Matrix

The aim of the Matrix method is to group all trips starting on a given edge and finishing on a target edge, by their lengths, L_{trip}, and calculate the sum of the coefficients A_{trip} of those trips for each particular group, instead of calculating coefficients individually for each trip. Because the method is centred around taking powers of an adjacency matrix Q, the number of trips starting from x and ending on any edge are calculated simultaneously, which is an advantage should we wish to compute $G_{ij}(x, y, t)$ for all j, that is, the Green's function response at a point x on branch i

for an injection anywhere along the tree. Coefficients are encoded into the matrix Q, the sums of coefficients are computed at the same time as trips are counted.

We begin by defining the function $c_k^s \; : \mathcal{E} \to \mathbb{R},\; s \in \mathcal{E}$, as the sum of all coefficients A_{trip} which begin at point $x \in s$ and travel over k edges, finishing on a given edge g :

$$c_k^s(g) \;=\; \sum_{\substack{\text{trips}\\ x\to\ldots\to y\\ \text{in } k \text{ jumps}}} A_{\text{trip}}\,, \qquad x \in s,\; y \in g\;.$$

Because $\mathcal{E}$ is a totally-ordered and finite set, then $\mathbf{c}_k^s = \Big(c_k^s(e_1),\,\ldots\,,c_k^s(e_{|\mathcal{E}|})\Big) \in \mathbb{R}^{|\mathcal{E}|}$ can be thought of as a vector, where $e_i \in \mathcal{E}$ for $i = 1,\,\ldots\,,|\mathcal{E}|$. The ith element of the vector $\mathbf{c}_k^s$ corresponds to the sum of coefficients A_{trip} for all trips originating at x on s and ending along the ith edge e_i, having travelled over k edges. Thus, the vector $\mathbf{c}_1^s$ consists mostly of zeros, with a one only in the entry corresponding to the edge s. This is because the coefficient taken for starting at x and moving along s remains equal to 1, while all other moves are invalid in only one hop, and hence have coefficient 0.

We can now define a matrix $Q \in \mathbb{R}^{\mathcal{E}\times\mathcal{E}}$ such that

$$Q^k\,\mathbf{c}_1 = \mathbf{c}_{k+1}. \tag{4.13}$$

Q is a modified form of the edge-adjacency matrix. With $\mathcal{E}$ ordered, the indices of an edge-adjacency matrix correspond to the ordering of the edge set $\mathcal{E}$. Instead of a Boolean matrix, however, we define the elements (i, j) of Q as containing the coefficient taken in moving from edge j to edge i. The entries of Q can thus be computed from the morphology of the graph. If the jth entry corresponds to edge (u, v) and the ith entry to edge (v, w), then the entry Q_{ji} is the coefficient taken when moving from branch (u, v) to (v, w). In the general case, these numerical values must be determined for each entry. However, in the simplified case where the radii on all branches are equal and all nodes have degree $d = 1$ or $d = 3$, the matrix Q can be constructed according to

$$Q_{ji} = \begin{cases} -\frac{1}{3}, & \text{if } j = (u,v) \;\text{ and }\; i = (v,u),\ \text{where } v \text{ is a node of degree } d = 3,\\ \;\;1, & \text{if } j = (u,v) \;\text{ and }\; i = (v,u),\ \text{where } v \text{ is a closed terminal } (d = 1),\\ \;\;\frac{2}{3}, & \text{if } j = (u,v) \;\text{ and }\; i = (v,w),\ \text{where } u \neq w,\\ \;\;0, & \text{otherwise.} \end{cases} \tag{4.14}$$

Note that the above rules apply to the transpose of Q_{ij}.

4.6.3 Computing the Path Integral

Thus, knowing the matrix Q from the dendritic geometry, and the vector c_1^s from the starting edge s, it is possible to calculate $c_k^s(g)$, the sum of coefficients for all trips travelling up to $k_{\max}$ edges, from $x \in s$ to $y \in g$. However, by considering trips moving from x in one direction only, and arriving at y from only one direction, we have calculated the coefficients of just Class 1 trips. In order to find coefficients for the remaining three Classes, we must also compute $c_k^{s'}(g), c_k^s(g')$ and $c_k^{s'}(g')$. Then, using (4.13), the Green's function in (3.114) can therefore be written as

$$\begin{aligned} G_{ij}(x,y,t) &= \sum_{\substack{\text{trips}\\ x \text{ to } y}} A_{\text{trip}}\, G_\infty(L_{\text{trip}}, t) \\ &= \sum_{k=1}^{k_{\max}} \Bigg[\Big(Q^{k-1}\,\mathbf{c}_1^s\Big)_g\, G_\infty\Big(L_1(k), t\Big) \\ &\qquad + \Big(Q^{k-1}\,\mathbf{c}_1^{s'}\Big)_g\, G_\infty\Big(L_2(k), t\Big) \\ &\qquad + \Big(Q^{k-1}\,\mathbf{c}_1^{s}\Big)_{g'}\, G_\infty\Big(L_3(k), t\Big) \\ &\qquad + \Big(Q^{k-1}\,\mathbf{c}_1^{s'}\Big)_{g'}\, G_\infty\Big(L_4(k), t\Big) \Bigg], \end{aligned} \tag{4.15}$$

where $\big(Q\,\mathbf{c}_1^s\big)_g$ is the gth element of the matrix-vector product of Q and $\mathbf{c}_1^s$. Lengths $L_1, ..., L_4$ are the lengths of Class 1 to Class 4 trips, respectively, and are defined as

$$\begin{aligned} L_1(k) &= \Delta x\Big(2(k-1) + x + y\Big), \\ L_2(k) &= \Delta x\Big(2k - x + y\Big), \\ L_3(k) &= \Delta x\Big(2k + x - y\Big), \\ L_4(k) &= \Delta x\Big(2(k+1) - x - y\Big). \end{aligned}$$

By selecting a small Δx, branches may be well approximated by a discretisation using an integer number of edges of length Δx. As in compartmental models, this allows the full morphology of the dendritic tree to be approximated, in a trade-off between high speed (large Δx) and accuracy (small Δx). As $\Delta x \to 0$, however, this approach tends to the computational complexity of naively integrating the cable equation using numerical methods. As in numerical simulations, where reducing Δx in order to increase accuracy brings about a necessary and associated change in Δt, the same is true of the Matrix method : selecting a small Δx and hence, increasing

$|\mathcal{E}|$, implies that $k_{\max}$ must be increased.

In all cases with bounded node degree, Q is a sparse matrix with only a few entries per row, and $\mathcal{O}(|\mathcal{E}|)$ entries altogether, making the complexity for the calculation of all coefficients $\mathcal{O}(|\mathcal{E}|\, k_{\max})$ by using highly-efficient sparse linear algebra algorithms. This compares positively with the algorithms previously described : the Trip-Grouping Matrix method is linear in $k_{\max}$, related to the maximum length of the trips. As $k_{\max}$ increases, the total number of trips increases exponentially, and hence, an increase in maximum trip length yields a greater-than-linear number of trips counted and coefficients computed.

4.6.4 Example Calculation

Here, we demonstrate an example realisation of the Matrix method for a dendritic structure composed of three branches of equal length Δx and equal radius, shown in Figure 4.2. In this symmetrical case, the matrix Q is very small, and can be constructed by hand. We place the point of measurement x along edge $s = (A, B)$, and point of current injection y along $g = (B, D)$. We begin by ordering the edge pairs as follows :

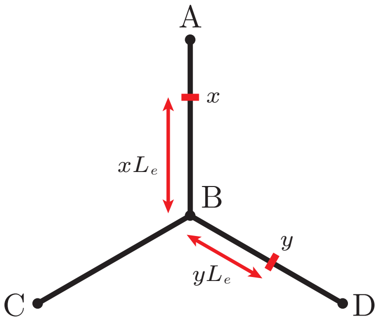


Figure 4.2: A model branching structure for the example calculation for the Matrix method. All three branches have length L and the same radius.

$$(A, B),\ (B, A),\ (B, C),\ (C, B),\ (B, D),\ (D, B).$$

Based on this ordered set, we can obtain two coefficients vectors : one for trips that begin at x and move towards B, denoted c_1^s, and one for trips moving from x towards node A, denoted $c_1^{s'}$. The first of these will help us generate Class 1 and Class 3 trips, while the latter will be used in the construction of Class 2 and Class 4 trips. These vectors are found to be

$$\mathbf{c}_1^s = \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T , \tag{4.16}$$

$$\mathbf{c}_1^{s'} = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}^T . \tag{4.17}$$

Using the rules described in (4.14), we can construct the matrix Q for our dendritic structure as follows :

$$Q = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 0 & 0 & \frac{2}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & 0 & 0 & -\frac{1}{3} & 0 & \frac{2}{3} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{2}{3} & 0 & 0 & \frac{2}{3} & 0 & -\frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} .$$

Note that all rows and columns sum to 1. The matrix Q is, in this way, somewhat analogous to the bistochastic transition matrix for a Markov chain. Instead of representing transition probabilities, however, its elements represent the cost of taking a particular transition to the overall Green's function solution. As such, the entries of Q are allowed to be negative.

Knowing this matrix Q and breaking the trips into the four main classes, it is straightforward to find the complete Green's function :

$$\begin{aligned} G_{sg}(x,y,t) = &\sum_{k=1}^{k_{\max}} \Big(Q^{k-1}\,\mathbf{c}_1^s\Big)_g\, G_\infty\Big(\Delta x\big(2(k-1)+x+y\big),t\Big) \\ &+\sum_{k=1}^{k_{\max}} \Big(Q^{k-1}\,\mathbf{c}_1^{s'}\Big)_g\, G_\infty\Big(\Delta x\big(2k-x+y\big),t\Big) \\ &+\sum_{k=1}^{k_{\max}} \Big(Q^{k-1}\,\mathbf{c}_1^s\Big)_{g'}\, G_\infty\Big(\Delta x\big(2k+x-y\big),t\Big) \\ &+\sum_{k=1}^{k_{\max}} \Big(Q^{k-1}\,\mathbf{c}_1^{s'}\Big)_{g'}\, G_\infty\Big(\Delta x\big(2(k+1)-x-y\big),t\Big). \end{aligned} \tag{4.18}$$

By summing over the number of hops allowed, k, each time taking the product between a sparse matrix and a vector, the Green's function $G_{sg}(x,y,t)$ is computed for injection at y on edge g and measurement at x on edge s. The matrix-vector product $\big(Q^{k-1}\mathbf{c}_1\big)$ actually yields results for the sum of coefficients for trips finishing on all edges; the selection of a given element g in the matrix-vector product in (4.18) allows the calculation of the Green's function for injection at a given location, but the solution G_{sg} is computed for all $g \in \mathcal{E}$ simultaneously, offering considerable computational efficiency.

Conclusions

The Improved Four Classes, the Length Priority and the Monte Carlo methods all construct trips explicitly, and one at a time. They are therefore constrained to a linear lower bound on scaling, with the number of trips constructed. The Trip-Grouping Matrix algorithm discretises the dendritic tree in order to group all trips of equal length into one iteration of the algorithm. For sparse discretisations, this yields extreme efficiency, at a cost to accuracy. This method is analogous to a generalisation of our solution for symmetrical star graphs, presented in Section 3.6.2.

4.7 Convergence of Time-Domain Methods

All of the algorithms for approximating the Green's function in the time-domain described in this chapter consist of infinite sums. Whilst the series solution for the dendritic path integral is proven to converge for all finite acyclic geometries with bounded degree, the convergence analysis in Section 3.5.3 suggests an error that scales with e^{-N} for a series solution including all trips up to N nodes in length, assuming an *optimally-ordered* sum. The Four Classes, Length Priority and Trip-Grouping algorithms use length heuristics to order the terms, while the Monte Carlo method generates randomly-ordered terms. Therefore, none of these methods construct a sum in which the terms are optimally-ordered, and we must differentiate between the theoretical convergence of the dendritic path integral and the computational convergence of the various algorithms which implement it.

4.7.1 Morphologies

Calculations of the Green's function (3.114) were made on a number of branching structures taken from the digital reconstructions of real neurons published in peer-reviewed literature. The NeuroMorpho Database [Ascoli *et al.*, 2007] hosts a large number of such reconstructions in `.swc` format. These files describe a sequence of nodes with precise radii and three-dimensional locations to describe the location of the soma and the paths taken by the axon and each dendrite. A dendrite's path can be described using as many nodes as necessary to accurately reflect the spatial jitter and variation in radius of its path. Because radii are described at nodes, edges between two nodes of different radius taper. The dendritic path integral formalism requires constant diameter along edges, but allows discontinuous jumps in the diameters at nodes. Hence, edge diameter was defined as the average of the diameters of adjacent nodes. This allows full dendritic branches to be represented as a sequence of uniform cylinders of arbitrary length and with abrupt changes in diameters at nodes.

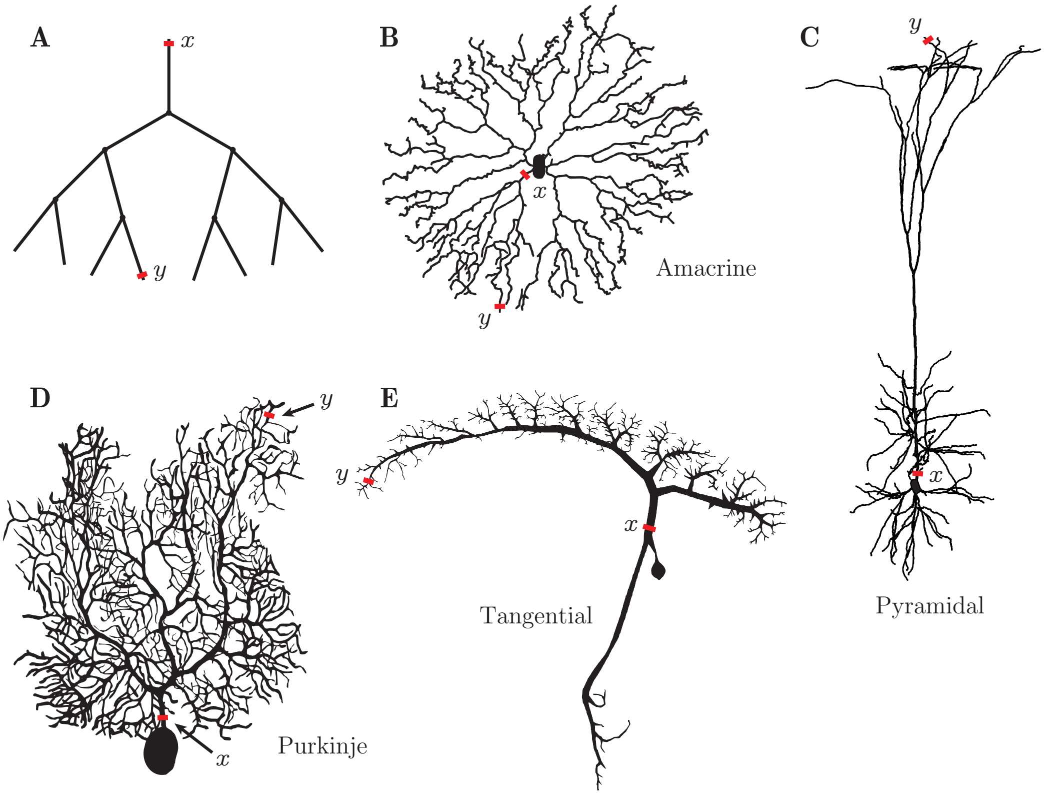


Figure 4.3: Neuronal structures used in construction of the Green's function. **A** : a fourth-order binary tree, **B** : a rabbit amacrine cell [Bloomfield and Miller, 1986], **C** : a rat pyramidal cell [Radman *et al.*, 2009], **D** : a rat Purkinje cell [Vetter *et al.*, 2001], and **E** : a blowfly tangential cell [Cuntz *et al.*, 2008].

The information contained within these `.swc` files required processing before it could be used by any of the algorithms described in this section. A parser was therefore written in Python, to provide the preliminary data processing to construct an input object capable of being passed to the algorithm, such as an adjacency list for the Monte Carlo method and the Length Priority algorithm.

Figure 4.3 shows the neuronal geometries used in the validation and convergence analysis of these algorithms. Figure 4.3A is a fourth-order binary tree where all branches have equal length and radius, morphologically close to the binary tree in Cao and Abbott's [1993] Figure 2B. The structures in Figures 4.3B-E are reconstructions from micrographs for four different neuronal types with qualitatively very different dendritic trees. For calculations run on the binary tree, three different sets of parameters were used, as shown in Table 4.1. Parameters in Set A are taken from Cao and Abbott [1993], while Sets B and C are more biologically realistic parameters.

For each morphology, points x and y were selected such that the measurement point, x, was adjacent to the soma; this situation is interesting should we wish the assess the somatic impact of a postsynaptic stimulation. The injection point, y, was placed as far from the soma as possible, to simulate the input of a distal postsynaptic potential.

4.7.2 Implementations

The Improved Four Classes Algorithm

The Improved Four Classes algorithm was written purely in C++, using containers and sorting algorithms from the Standard Template Library.

The Length Priority Algorithm

The Length Priority algorithm was implemented primarily in C with an implementation of the optimised Eppstein [1999] algorithm provided by Jiménez and Marzal [2003]. From this, the lengths and coefficients of the trips, and subsequently, the Green's functions, were computed in C++.

The Monte Carlo Method

The Monte Carlo method can easily be parallelised, as all trip realisations are generated independently. By also providing a natural means for string manipulation for the discovery of subwalks, Matlab [2012] was a natural choice for the implementation of the Monte Carlo method.

The Trip-Grouping Matrix Method

The Matrix method is centered around sparse matrix-vector multiplications. As such, Matlab [2012] was used in its implementation.

	Parameter Set A	Parameter Set B	Parameter Set C
Branch length L	0.3	50 μm	100 μm
Branch diameter a	0.05	1 μm	1 μm
Diffusion coefficient D	1	2.5 $\times 10^4$ μm^2 ms^{-1}	2.5 $\times$ 10^4 μm^2 ms^{-1}
Time constant τ	1	3.3 ms	3.3 ms
Capacitance C_{m}	1	1 μF cm^{-2}	1 μF cm^{-2}

Table 4.1: Parameter sets of the binary tree in Figure 4.3A.

4.7.3 Validation Against Numerics

We first validate the computational implementations of our algorithms by constructing the Green's function $G_{ij}(x, y, t)$ on a small binary tree. Two profiles of the response function obtained by the Length Priority, the Monte Carlo and the Trip-Grouping Matrix methods, compared to a numerical simulation computed by the software package NEURON [Carnevale and Hines, 2006], are shown in Figure 4.4. NEURON can used as a reference when configured such that the dendritic trees are discretised into very small segments, and the time step is also small. These simulations tend to run very slowly, but this configuration of parameters ensures that the accumulated error remains small. For large networks of trees, however, this would be infeasible due to time constraints, and one of the methods proposed here may be more suited to large-scale simulations.

These plots demonstrate an excellent agreement between the different algorithms and the numerical solution, with a slightly worse performance of the Monte Carlo method for larger times. Due to the process of extracting subwalks from longer walks of fixed length, then long trips are undersampled in comparison with shorter trips, and so convergence at long times requires more sampling.

4.7.4 Error of Convergence

The error made during the calculation of the Green's function (3.114) is a direct measure of the how far the computation is from convergence. We use the following

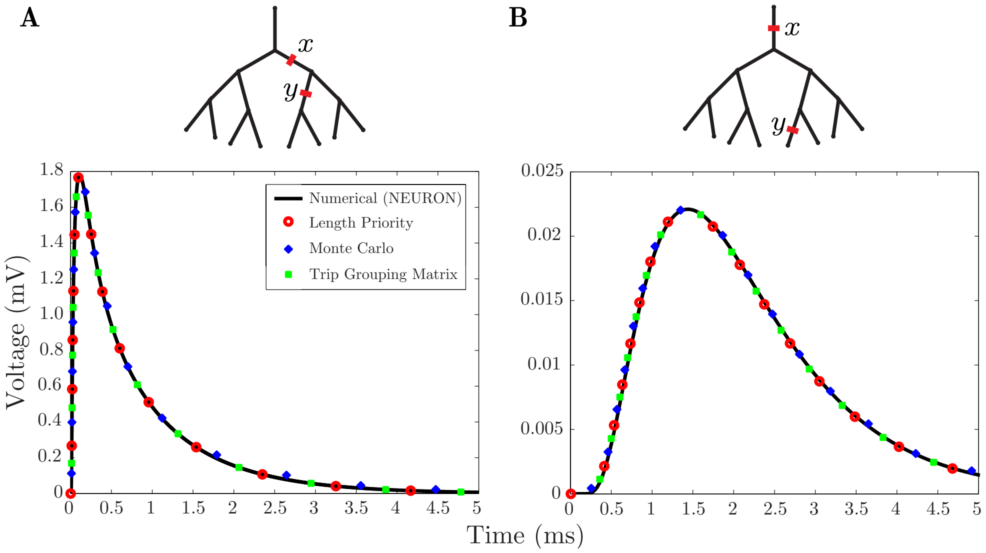


Figure 4.4: The Green's function constructed by a number of methods. The voltage traces show the computed solutions $G_{ij}(x, y, t)$ for fixed x and y on a binary tree for the Length Priority method (red circles), the Monte Carlo method (blue diamonds) and the Trip-Grouping Matrix method with $k_{\max} = 100$ (green squares) superimposed on NEURON's numerical solution (black line). Parameter set A (Table 4.1) was used in these computations.

normalised L^1 error as a measure of convergence :

$$\varepsilon = \frac{1}{V_N} \int\limits_0^T \Big| G_{ij}(x, y, t) - V^*(x, y, t) \Big| \, \mathrm{d}t, \tag{4.19}$$

where $T = 20$ ms is the final simulation time, $G_{ij}(x, y, t)$ is calculated algorithmically using one of the methods described in this chapter, $V^*(x, y, t)$ is NEURON's numerical solution to very high accuracy, and

$$V_N = \int\limits_0^T V^*(x, y, t) \, \mathrm{d}t \tag{4.20}$$

is the integral of the accurate NEURON solution. This convergence measure is therefore relative to the amplitude of the "real" solution, and thus errors ε are comparable between different neuronal types.

4.7.5 Convergence of the Length Priority Methods

Figure 4.5 shows the error of convergence (4.19) of the Improved Four Classes and of the Length Priority methods on the five geometries from Figure 4.3, as a function of the number of trips generated. Three sets of parameters given in Table 4.1 were considered for the binary tree in Figure 4.3A, and the relative errors ε for each case are demonstrated in Figures 4.5A-C. These plots illustrate fairly uniform convergence, in which both methods offer similar accuracies and rates of convergence. Of the two trees with biophysically realistic parameters, the binary tree with the longer branches (Parameter Set C) converges faster, as is expected on structures with longer trips in each Class. This is reflected in Table 4.2, which shows the number of trips required on the binary tree to remain under a given error threshold for different parameter sets.

Figures 4.5D-G show ε for the structures in Figures 4.3B-E respectively. They demonstrate that convergence is non-trivial on complex branching structures. Figure 4.5D shows that the Length Priority method makes consistently less error on the amacrine cell geometry, in contrast to the convergence of the Purkinje cell, shown in Figure 4.5E, where the Improved Four Classes method generates less error for all numbers of trips. Both of these show strongly irregular convergence, and high-amplitude oscillation in the errors ε in the amacrine cell. For both methods, the Purkinje cell shows a plateau in error for Green's functions with few trips, indicating that these trips either are of small magnitude, or that their voltage traces alternate between undershooting and overshooting the correct solution between subsequent trips. This indicates that neither the Length Priority nor the Improved Four Classes methods are necessarily good heuristics for ordering terms in the Green's function. This is further demonstrated at by the oscillating property of the error, which shows how there are regions where trips that will increase the error are more frequent than trips that reduce it.

The pyramidal cell's convergence shows very discontinuous behaviour (Figure 4.5F), particularly in the Length Priority method. The large jump in error when approximately 350 trips are included in the Green's function was found to be caused by the first and shortest Class 2 trip included thus far, with all prior trips belonging to Class 1. This behaviour is likely to arise if there exist very short branches along the shortest and most direct $x \to y$ trip, and thus many Class 1 trips are generated first, being shorter than the first Class 2 trip. Whilst one of the motivating reasons for considering a Length Priority approach was to generate trips fully by length order, this heuristic makes no attempt to include the coefficient A_{trip} in its ordering. This is an example of a pathologically large change in the coefficients value for a

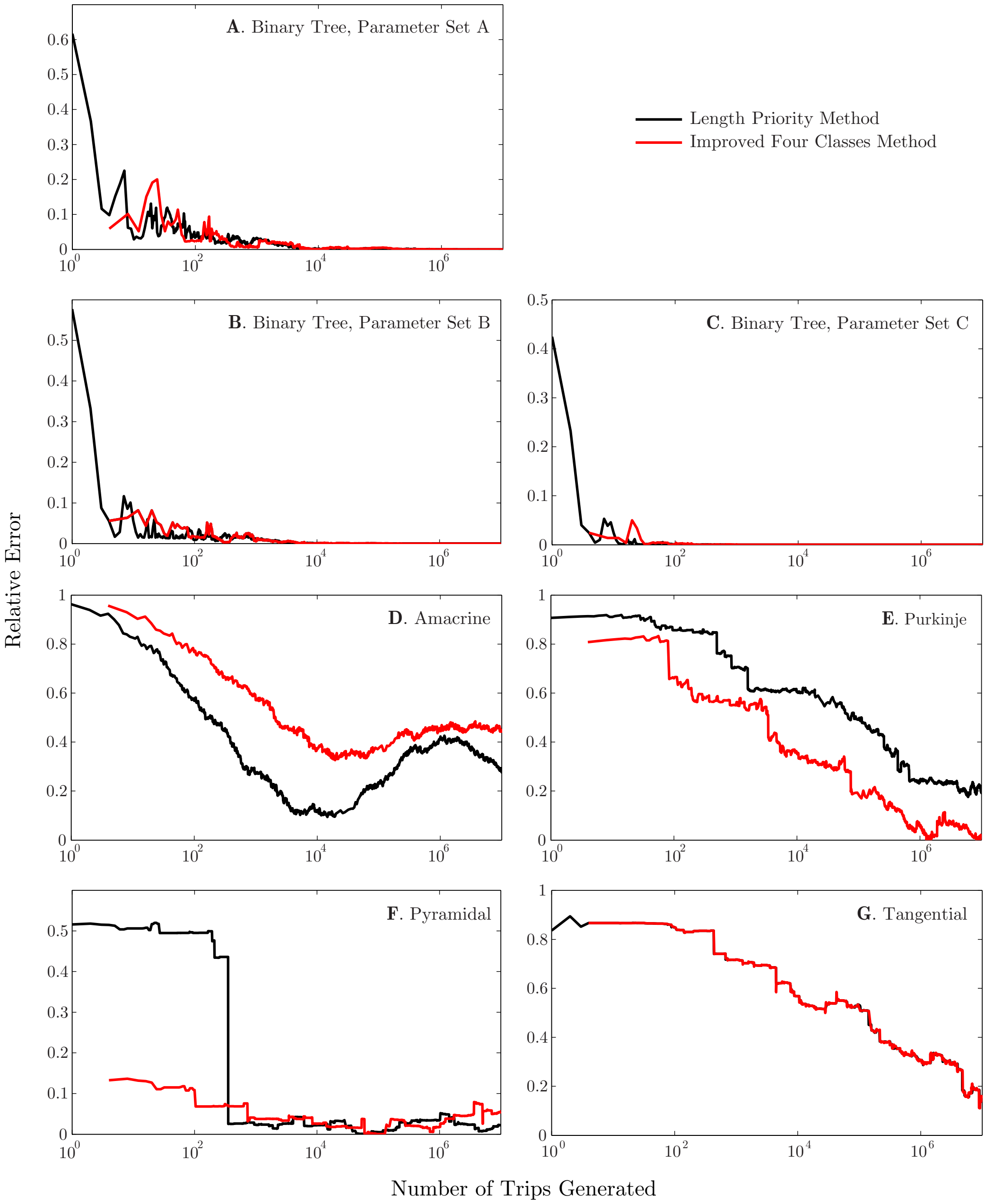


Figure 4.5: Convergence of the Improved Four Classes and Length Priority methods for a number of dendritic morphologies. The relative error ε of the approximation of $G_{ij}(x, y, t)$ is shown as a function of the number of trips in the sum-over-trips framework for injection at y and measurement at x on the dendritic trees in Figure 4.3. The membrane parameters for real dendritic morphologies are : $C_\text{m} = 1\ \mu\text{F cm}^{-2}$, $R = 3000\ \Omega\ \text{cm}^2$ and $R_a = 100\ \Omega$ cm. Note that the Improved Four Classes method always begins with four trips, and each iteration in the algorithm adds a further four trips, with one belonging to each class.

Binary tree	Relative Error Threshold ε			
	0.1	0.05	0.01	0.001
Parameter Set A	3240	8750	1820000	$> 5 \times 10^7$
Parameter Set B	825	2600	129000	$> 5 \times 10^7$
Parameter Set C	22	65	815	7700

Table 4.2: Length Priority method on a binary tree: number of trips required for a given accuracy.

Class 2 trip which contributes a very significant amount to the Green's function. The Improved Four Classes approach, which enforces generation of trips of all four classes at every added excursion, does not show such a drastic drop in error. However, the error plot is still very discontinuous, and this may be a characteristic of situations as we have just described, where points x and y are placed on branches having a very different length to those on the most direct $x \to y$ trip, or when these points are placed very close to a node. Whether injection and measurement points are located on branches that are significantly longer or shorter than those along the shortest $x \to y$ trip, both the Improved Four Classes and the Length Priority methods will generate trips in an "unnatural" order, subsampling the trips where current will spread the most, but oversampling in areas of the tree with very short branches. This pathological feature may not be inherently present in the real neuronal morphology, but may have been created during digital reconstruction from slice image data, if, for example, a change of radius were found along the branch. Therefore, this pathology may not be representative of the neuronal geometry, but becomes a function of the reconstruction.

The tangential cell's convergence, shown in Figure 4.5G, shows almost identical errors for both the Improved Four Classes and the Length Priority methods, indicating that trips are generated in a similar order, regardless of method. Contrary to the example with the pyramidal cell, this behaviour is likely to occur when x and y are placed on branches that are significantly shorter than those that arise on the shortest $x \to y$ path, such that the Length Priority method returns trips of Class 1, 2, 3 and 4 in sequential order, as these increases in length are shorter than adding an excursion along the direct $x \to y$ trip.

These results clearly indicate that the convergence of the realisation of the sum-over-trip framework by either the Improved Four Classes or the Length Priority method strongly depends on a dendritic geometry. For real morphologies, the number of trips required quickly becomes very large to the point where guaranteeing convergence to within some small error threshold may become computationally expensive.

4.7.6 Convergence of the Monte Carlo Method

The convergence of the Monte Carlo method is shown in Figure 4.6 for the binary tree in Figure 4.3A with Parameter Set A, and for a larger binary tree of order 16 with the same parameters. Boundary effects can be seen on the smaller tree, where the convergence rate is slightly faster than that of a typical Monte Carlo integration, which is observed here for a larger tree. It is worth noting that the x-axis on this plot shows the number of random walks generated; however, due to the number of subwalks extracted from each random walk, the number of terms contributing to the Green's function can potentially be significantly different. For small trees, the number of subwalks the algorithm extracts is typically much larger than the number of random hopper realisations sampled; larger trees have trips that can "get lost" far from the points of interest, meaning that a considerable section of the realisation does not contribute to the sum-over-trips.

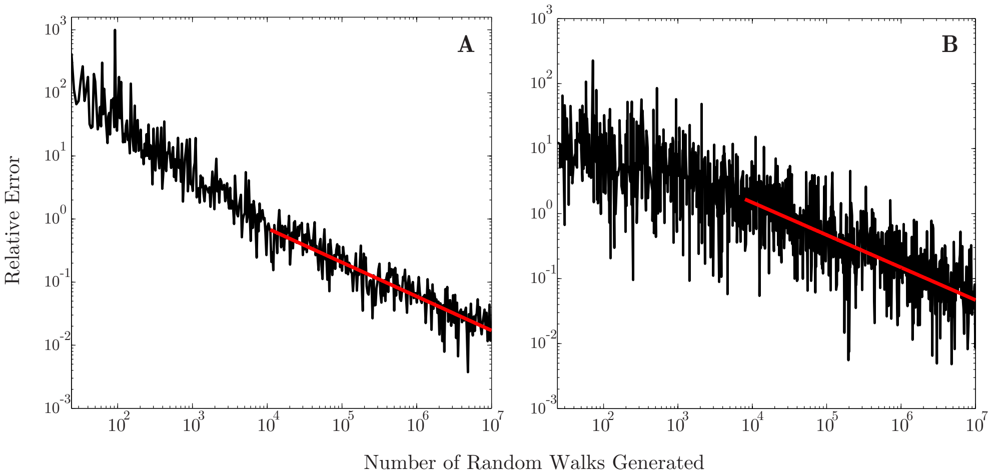


Figure 4.6: Convergence of the Monte Carlo method for a binary tree. The relative error ε is shown as a function of the number of random walk realisations generated, k. **A** : the error generated on the binary tree in Figure 4.3A with parameter set A. The red line shows a fit for $\varepsilon \sim k^{-0.54}$. **B** : the convergence error on a binary tree of order 16 (65536 nodes). The red line demonstrates a fit for $\varepsilon \sim k^{-0.5}$, the typical rate of convergence of a Monte Carlo integration.

The graphs in Figure 4.6 show that the Monte Carlo method is very slow to converge, although the method is much more predictable in its convergence, despite the noise. As expected, therefore, the Monte Carlo method generates trips that are more "naturally" ordered, and hence, convergence is much more monotonic. Despite this improved ordering of terms in the series solution, the Monte Carlo approach remains computationally intensive and very slow to converge with an increasing number of trips.

4.7.7 Convergence of the Trip-Grouping Matrix Algorithm

Finally, the convergence of the Matrix method on a binary tree is shown in Figure 4.7. This algorithm converges extremely quickly to within very small error tolerances as a function of $k_{\max}$, the maximum number of edges covered by trips generated. The values of the product of A_{trip} coefficients obtained by this method remain $\mathcal{O}(1)$ for all $k_{\max}$ which agrees with the proof in Abbott [1992].

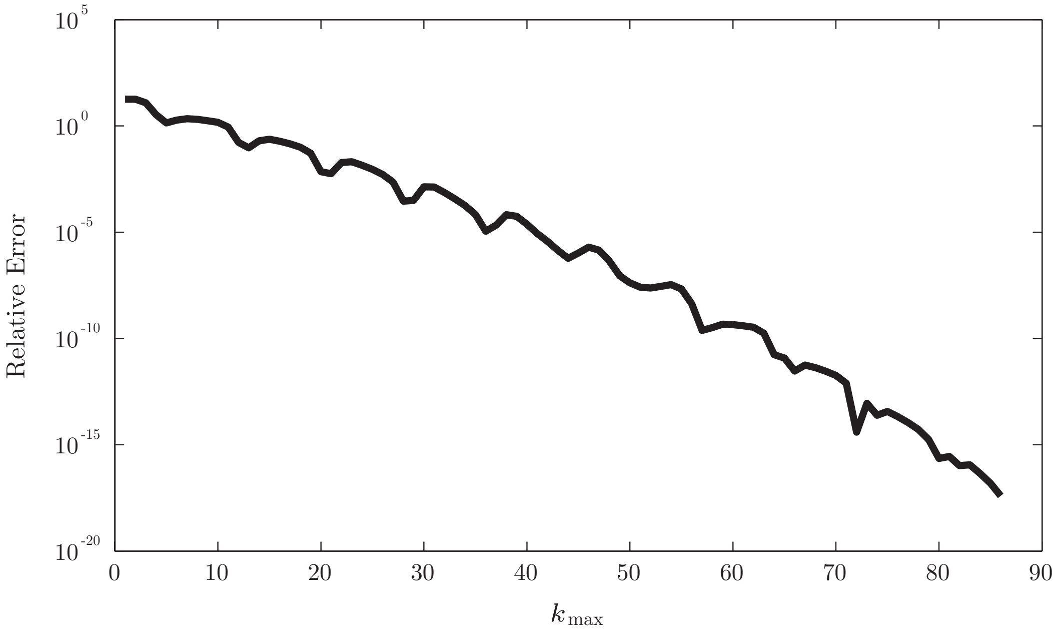


Figure 4.7: Convergence of the Trip-Grouping Matrix method. The relative error ε is shown as a function of the maximal number of edges travelled in the trips, $k_{\max}$, for a binary tree in Figure 4.3A.

Because the algorithm is based on simple matrix-vector multiplication, where the matrix is $|\mathcal{E}| \times |\mathcal{E}|$ in size, the computation of the Green's function for small trees such as the binary trees used here to within $\varepsilon = 10^{-15}$ only takes a fraction of a second on a desktop computer. On more complex trees such as Purkinje cells, this becomes more expensive, although computing $G_{ij}(x, y, t)$ for the whole tree (for all i and j) remains computationally preferable to the use of brute-force simulators. Using the reciprocal rule (3.116), this is possible in $|\mathcal{E}|/2$ applications of the algorithm. This compares favourably with the Length Priority and the Four Classes methods, which require $|\mathcal{E}|\big(|\mathcal{E}| + 1\big)/2$ applications of the algorithm, and with NEURON, which would require $|\mathcal{E}|^2$ simulations, and this would only provide solutions for a single point y on each edge. In addition, the sparseness of the matrix Q means that coefficient calculation up to $k_{\max}$ only takes $\mathcal{O}\big(|\mathcal{E}|\, k_{\max}\big)$ time, and so the method scales linearly with the number of branches on the tree.

Due to the Trip-Grouping Matrix method's extreme efficiency, we can use it to probe the convergence of the dendritic path integral as a function of system time, in addition to its convergence on different morphologies. In Figure 4.8A, we have constructed the Green's function for a depth-4 binary tree using the Trip-Grouping Matrix method, for times $0 \leq t \leq 50$ ms, and computed its relative error as in (4.19), except for the removal of the absolute value function, in order to better demonstrate the oscillatory property of the dendritic path integral's convergence. These results show how, whilst small errors may be made at short times, as agrees with Cao and Abbott [1993], they rapidly grow for longer times. Because of this, the number of trips chosen to compute in the path integral should be informed by the length of the simulation : for long-duration simulations, orders of magnitude more trips are required to obtain the same accuracy.

Figure 4.8B shows the same calculation for a depth-12 binary tree. It shows that, comparatively, it is much harder to achieve convergence on a larger tree. Even at short times, errors are present for a large number of trips, and, for longer times, we could require as many as 10^{30} trips on a tree of this size - a number which is infeasible with regards to explicit trip construction, for modern computing resources. This places trip-enumerating methods such as the Trip-Grouping Matrix method at a significant advantage, compared with those that construct trips explicitly : the prior are able to evaluate a far greater number of trips than the latter. This is evident in comparing Figures 4.5 and 4.6, where we constructed 10^7 trips, with Figure 4.8, where we were able to obtain 10^{36} trips, in a similar amount of computation time.

4.7.8 Structural-Electrotonic Properties

The Green's function $G_{ij}(x, y, t)$ for a given dendritic geometry, being equivalent to an impulse response function, can be used to assess the efficacy with which electric signals are transmitted on a given dendritic morphology. Assessing whether a the geometry of a branching structure significantly impacts the propagation of a signal

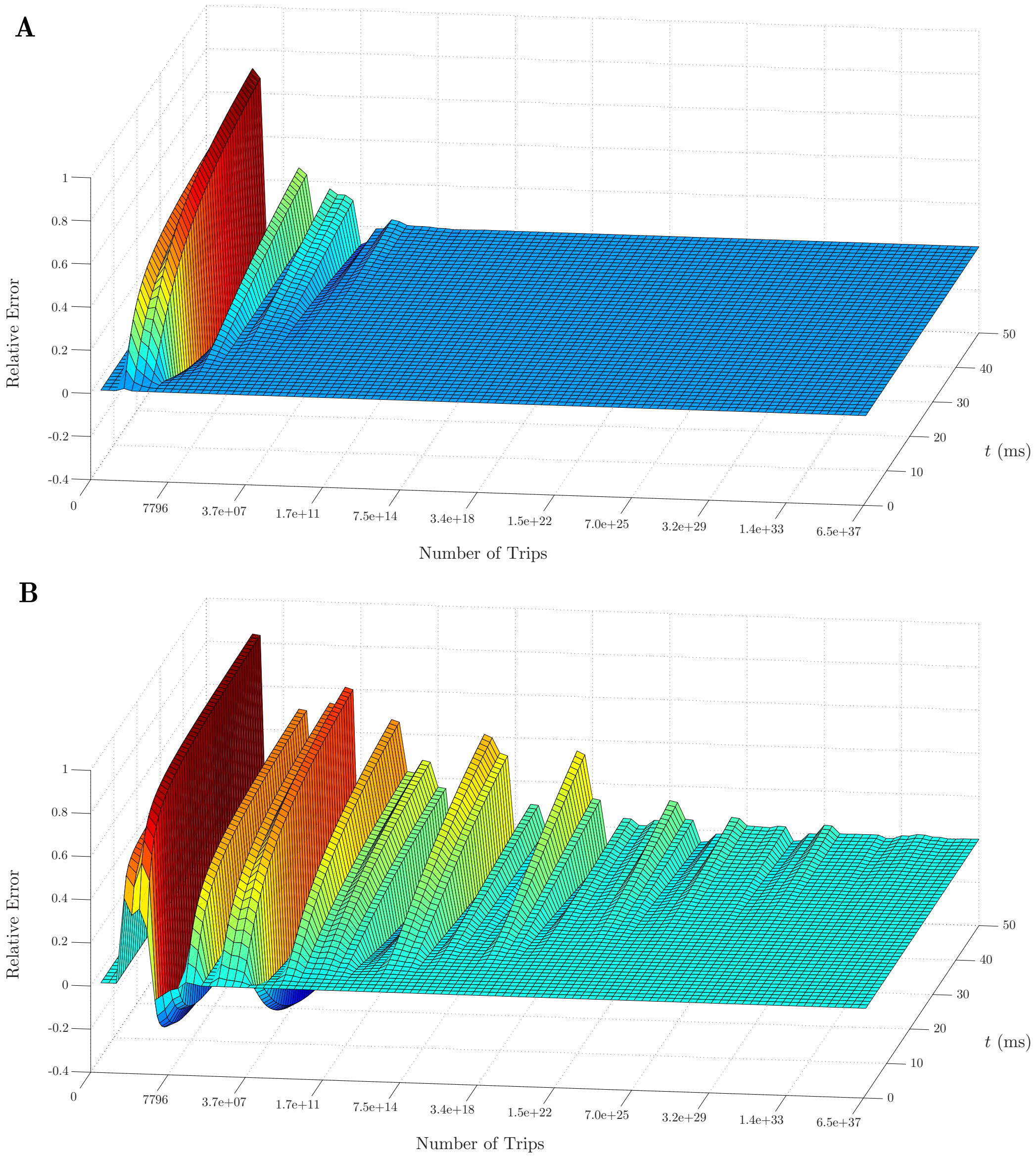


Figure 4.8: Convergence error, as a function of number of trips and simulation time.

is a step towards answering questions regarding the structure-function relationship behind the enormous natural variation in dendritic morphologies. A dendritic tree's electrotonic properties could hint at the role of the neuron in terms its signal integration and processing of information.

Using the measures introduced by Zador *et al.* [1995], the propagation delay and the log-attenuation, we can analyse the how the variation in dendritic morphology affects the propagation of the response signal in the four reconstructed cells in

Figure 4.3, effectively mapping a measure of structure to one of function. For a pair of points (x, y) along the tree, we reintroduce the propagation delay $\mathcal{P}_{xy}$ from Zador *et al.* [1995] as a measure of the impact of the tree's electrotonic structure on the timing of signals, defined as

$$\mathcal{P}_{xy} = \hat{t}_x - \hat{t}_y. \tag{4.21}$$

Here, $\hat{t}_x$ and $\hat{t}_y$ are the geometric centroids of the voltage transients measured at x on branch i and at y on branch j, respectively, for a stimulus at y on j :

$$\hat{t}_x = \frac{\int\limits_0^\infty t\, G_{ij}(x, y, t)\, \mathrm{d}t}{\int\limits_0^\infty G_{ij}(x, y, t)\, \mathrm{d}t} \qquad \text{and} \qquad \hat{t}_y = \frac{\int\limits_0^\infty t\, G_{jj}(y, y, t)\, \mathrm{d}t}{\int\limits_0^\infty G_{jj}(y, y, t)\, \mathrm{d}t}. \tag{4.22}$$

The propagation delay $\mathcal{P}_{xy}$ is a measure of the speed at which signals are able to diffuse along the dendritic tree from the point at which current is injected. Should we use the Green's function $G_\infty(x, y, t)$ as the signal itself, the centroid for the response at the point of injection $\hat{t}_y$ can be found in closed form. First, we rewrite the numerator as a Gamma function :

$$\begin{aligned}\int\limits_0^\infty t\, G_{jj}(y, y, t)\, \mathrm{d}t \;&=\; \int\limits_0^\infty t\, \frac{1}{\sqrt{4\pi D t}}\, \mathrm{e}^{-t/\tau}\, \mathrm{d}t \\ &= \frac{1}{\sqrt{4\pi D}} \int\limits_0^\infty \frac{t}{\sqrt{t}}\, \mathrm{e}^{-t/\tau}\, \mathrm{d}t.\end{aligned} \tag{4.23}$$

Introducing $z = t/\tau$ such that $\mathrm{d}t = \tau\, \mathrm{d}z$, then we can write (4.23) as

$$\begin{aligned}\frac{1}{\sqrt{4\pi D}} \int\limits_0^\infty \frac{\tau z}{\sqrt{\tau z}}\, \mathrm{e}^{-z}\, \tau\, \mathrm{d}z \;&=\; \frac{\tau^2}{\sqrt{4\pi D \tau}} \int\limits_0^\infty z^{1/2}\, \mathrm{e}^{-z}\, \mathrm{d}z \\ &= \frac{\tau^2}{\sqrt{4\pi D \tau}}\, \Gamma(3/2),\end{aligned} \tag{4.24}$$

where

$$\Gamma(x) = \int\limits_0^\infty z^{x-1}\, \mathrm{e}^{-z}\, \mathrm{d}z \tag{4.25}$$

is the generalised Gamma function. Similarly for the denominator, we find that

$$\begin{aligned}\int_0^\infty G_{jj}(y,y,t)\ \mathrm{d}t &= \frac{1}{\sqrt{4\pi D}} \int_0^\infty t^{-1/2}\ \mathrm{e}^{-t/\tau}\ \mathrm{d}t \\ &= \frac{\tau}{\sqrt{4\pi D\tau}} \int_0^\infty z^{-1/2}\ \mathrm{e}^{-z}\ \mathrm{d}z \\ &= \frac{\tau}{\sqrt{4\pi D\tau}}\ \Gamma(1/2). \end{aligned} \tag{4.26}$$

Reassembling the fraction in 4.22, we find that

$$\begin{aligned}\hat{t}_y &= \tau\ \frac{\Gamma(3/2)}{\Gamma(1/2)} \\ &= \frac{\tau}{2}. \end{aligned} \tag{4.27}$$

The propagation delay can thus be written simply, as

$$\mathcal{P}_{xy} = \frac{\int_0^\infty t\ G_{ij}(x,y,t)\ \mathrm{d}t}{\int_0^\infty G_{ij}(x,y,t)\ \mathrm{d}t} - \frac{\tau}{2}. \tag{4.28}$$

This delay measure admits an additive property, such that $\mathcal{P}_{xy} = \mathcal{P}_{xz} + \mathcal{P}_{zy}$ for a point z between x and y.

The other of Zador *et al.*'s [1995] measure of electrotonic structure, the log-attenuation of the response signal between a pair of points (x, y), is defined as

$$\mathcal{A}_{xy} = \log\left[\frac{\int_0^\infty G_{jj}(y,y,t)\ \mathrm{d}t}{\int_0^\infty G_{ij}(x,y,t)\ \mathrm{d}t}\right]. \tag{4.29}$$

It acts as a measure of the amount that a transient signal's amplitude diminishes as it travels between two points. $\mathcal{A}_{xy}$ is also additive for a point z between x and y, that is, $\mathcal{A}_{xy} = \mathcal{A}_{xz} + \mathcal{A}_{zy}$.

Using Zador *et al.*'s [1995] two measures, we can compute the Green's func-

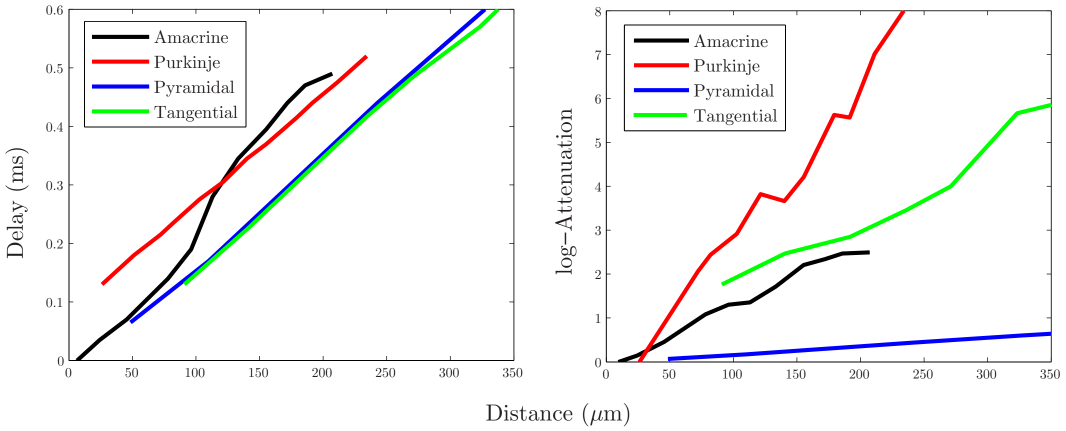


Figure 4.9: Propagation delay and log-attenuation for reconstructed geometries in Figure 4.3.

tion for different reconstructed dendritic geometries, and compare how the morphology impacts signal attenuation and delay. Figure 4.9 demonstrates the propagation delay and the log-attenuation as a function of distance away from a fixed point of measurement x, for the reconstructed dendritic morphologies in Figure 4.3B-E. The point x was placed near the soma, as in this figure; the position of y was moved away from x to the distal dendrites along a single path, finishing in the same position y as shown in the figure. As expected with an additive property, both the delay and the log-attenuation are linear in the distance between y and x. The curves in Figure 4.9 show a noisy linear trend, which could be smoothed to better demonstrate this linearity by sampling the data from multiple points located along different branches, but at the same fixed distance away from x, in the manner of an expanding sphere of radius y, with origin at x (where an average must be taken for x everywhere along the tree).

To compare how the response signal is transferred in four neuronal types, we plot the rate of change of the delay and log-attenuation for individual cells in Figure 4.10, computed by a linear regression and imposing that the line passes through the origin. It succinctly illustrates that the input signal will spread differently in these four cells, with the tangential cell having a similar rate of delay as the pyramidal cell and a similar log-attenuation as the amacrine cell. The signal in the Purkinje cell is shown to be attenuated most, whereas the pyramidal cell transfers signals very effectively. Similar conclusions, but about the propagation of the dendritic action potential, were made by Vetter *et al.* [2001]. The activation of voltage-gated channels is expected to be less robust in the case of strong attenuation of the passive spread of voltage which might explain the results of Vetter *et al.* [2001].

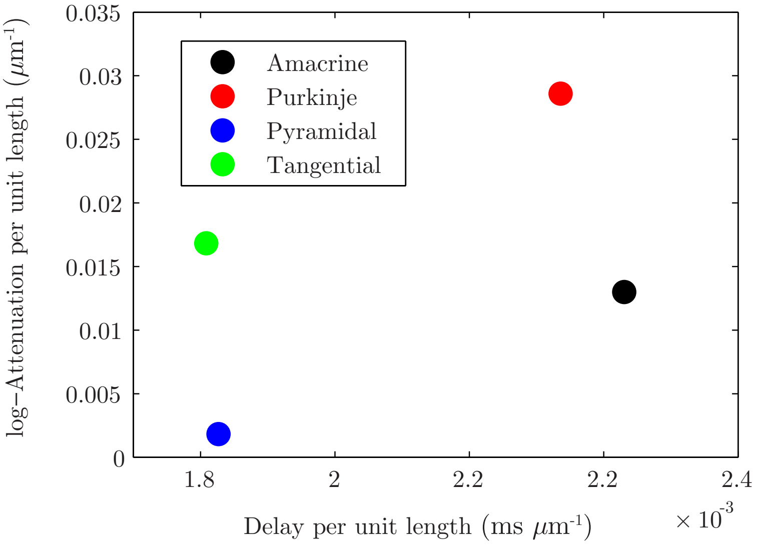


Figure 4.10: Electrotonic properties of the response signal for reconstructed geometries.

4.8 Conclusions

The algorithms presented in this chapter, and first published in Caudron *et al.* [2012], implement Abbott *et al.*'s [1991] dendritic path integral framework for computing the impulse response function for a dendritic tree directly in the time domain. The solution is constructed as an infinite sum which, in theory, converges exponentially quickly in the number of terms in the sum. The major difficulty is constructing the sum in an efficient order, where the absolute value of each term contributes more than the that of the next term. This is illustrated in the convergence analysis of the algorithms described here, which show a highly non-monotonic convergence for terms that are ordered by their lengths, and a much slower convergence for the Monte Carlo method where terms are randomly sampled.

Nevertheless, the series solution is guaranteed to converge, and therefore, with enough computational power, a solution can be found to within acceptable error tolerances. The Trip-Grouping algorithm proposed is especially rapid, producing very small errors in extremely short times, although it requires the tree to be discretised as in numerical simulation.

The resulting Green's function allows us to gain insight into how the dendritic tree's morphology affects its signal propagation efficacy. We have demonstrated how to calculate the extent to which a diffusing electrical signal is delayed as a function of the distance it must travel, and how much it will be attenuated in doing so.

These time-domain methods represent a powerful framework for investigating the voltage response to a current injection on dendritic trees with arbitrary morphology. Unlike numerical methods used in compartmental modelling, these methods scale favourably with system size and could be used to tackle large spatially-extended systems of dendritic trees. Bypassing the requirement for a numerical inverse Laplace transform, inherent to the vast majority of cable theory methods, is a strong incentive for use of the dendritic path integral. However, the solution remains an approximation, and significant computational power is required to construct it. In contrast, Laplace-domain methods can provide an exact solution, with error only being introduced during the numerical inverse transform. The next chapter will therefore introduce a novel method for constructing the Green's function in the Laplace domain, with a view to attacking cable problems from another angle.

Chapter 5

Laplace-Domain Methods

With the cable equation taking a simpler form in the Laplace domain, it is easy to see why methods rooted in the frequency domain outnumber those that operate in the time domain. Indeed, many Laplace-domain approaches, such as those of Butz and Cowan [1974], Koch and Poggio [1985] and Holmes [1986] provide exact solutions to the cable equation in the Laplace domain. In contrast, Abbott *et al.*'s [1991] dendritic path integral delivers only an approximation, and one which is not always trivial to obtain to the desired accuracy, as demonstrated in the previous chapter.

The method presented in the following sections revolves around the use of *motifs*, or minimal substructures extracted from the dendritic morphology, to put together a system of equations which can then used to solve the Laplace-domain solution to the cable equation. A major advantage of the motif idea is that the algorithm is very easy to apply, as each motif, identified from a node on the tree, leads to the concatenation of a small matrix into a larger system of equations.

Before introducing the algorithm that makes use of motifs, we review the Laplace-domain solution to the cable equation and introduce an alternate form in Section 5.1. Motifs are formally introduced in Section 5.2, and applied to an algorithm in Section 5.3.

5.1 Cable Systems in the Laplace Domain

For an arbitrary tree with $\mathcal{V}$ nodes and $\mathcal{E}$ branches, we can consider a system of $|\mathcal{E}|$ equations as in (3.81), with one for each branch $i \in \mathcal{E}$:

$$\begin{aligned} \frac{\partial V_i(x,t)}{\partial t} &= D_i \frac{\partial^2 V_i(x,t)}{\partial x^2} - \frac{V_i(x,t)}{\tau_i}, && i \neq j, \quad 0 \leq x \leq L_i, \\ \frac{\partial V_j(x,t)}{\partial t} &= D_j \frac{\partial^2 V_j(x,t)}{\partial x^2} - \frac{V_j(x,t)}{\tau_j} + I_{\text{app}}(x,t), && 0 \leq x \leq L_j, \end{aligned} \tag{5.1}$$

for $t \geq 0$ and $i, j \in \mathcal{E}$, and where the indices in V_i indicate the transmembrane potential on branch i. Each branch is allowed its own lengthscale λ_i and characteristic time τ_i, encoded in the space constant $D_i = \lambda_i / \tau_i$.

The current $I_{\text{app}}(x,t)$ is injected only along branch j. As in Section 3.4.5, and specifically in (3.89), we will set $I_{\text{app}}(x,t)$ to the delta function $\delta(x-y)\,\delta(t)$, in order to solve for the system's Green's function for injection at an arbitrary point y on branch j at time $t = 0$, without loss of generality.

In the Laplace domain, the Green's function for the system described in (5.1) can be written in a form similar to equation (3.90) :

$$\begin{aligned} \mathcal{L}\left[\frac{\partial G_i(x,\,\cdot\,)}{\partial t} - D_i \frac{\partial^2 G_i(x,\,\cdot\,)}{\partial x^2} + \frac{G_i(x,\,\cdot\,)}{\tau_i} \right](s) &= 0, \qquad i \neq j, \\ \mathcal{L}\left[\frac{\partial G_j(x,\,\cdot\,)}{\partial t} - D_j \frac{\partial^2 G_j(x,\,\cdot\,)}{\partial x^2} + \frac{G_j(x,\,\cdot\,)}{\tau_j} \right](s) &= \mathcal{L}\Big[\delta(x-y)\,\delta(\,\cdot\,)\Big](s), \end{aligned} \tag{5.2}$$

and therefore, assuming zero initial conditions as in Section 3.4.5, we find that

$$\begin{aligned} \gamma_i^2(s)\,\bar{G}_i(x,s) &= \frac{\partial^2 \bar{G}_i(x,s)}{\partial x^2}, \qquad i \neq j, \\ \gamma_j^2(s)\,\bar{G}_j(x,s) &= \frac{\partial^2 \bar{G}_j(x,s)}{\partial x^2} + \frac{\delta(x-y)}{D_j}, \end{aligned} \tag{5.3}$$

where

$$\gamma_i^2(s) = \frac{s + \tau_i^{-1}}{D_i}. \tag{5.4}$$

5.1.1 Laplace-Domain Solutions

Solutions to the set of equations (5.3) are found to be

$$\begin{aligned} \bar{G}_i(x,s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x}, \qquad i \neq j, \\ \bar{G}_j(x,s) &= \alpha_j(s)\,\mathrm{e}^{-\gamma_j(s)\,x} + \beta_j(s)\,\mathrm{e}^{\gamma_j(s)\,x} - I^s_j(x,y,s), \end{aligned} \tag{5.5}$$

as in [Boyce and DiPrima, 2012], and where the injection term $I^s_j(x,y,s)$ and its derivative $I^c_j(x,y,s)$, used later, are defined as

$$I^s_j(x,y,s) = \begin{cases} \dfrac{\sinh\big(\gamma_j(s)\,(x-y)\big)}{D_j\gamma_j(s)} & x > y, \\ 0 & \text{otherwise,} \end{cases} \tag{5.6}$$

$$I^c_j(x,y,s) = \frac{\partial I^s_j(x,y,s)}{\partial x} = \begin{cases} \dfrac{\cosh\big(\gamma_j(s)\,(x-y)\big)}{D_j\gamma_j} & x > y, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

for some fixed $0 > y > L_j$.

5.1.2 Boundary Conditions

With the system of equations in place, a set of boundary conditions must be considered at branching nodes and terminals. These are the Laplace-domain analogues of those covered in the description of the dendritic path integral in Section 3.5 : continuity of potential at nodes, as in (3.106), and Kirchhoff's current law, imposing that the sum of currents flowing into a node is equal to the sum of currents flowing out of it, as in (3.107). These must, however, be expressed in the Laplace domain, where continuity of potential imposes

$$\bar{G}_i(x_{\text{node}},s) = \bar{G}_j(x_{\text{node}},s), \tag{5.8}$$

at a node, for all pairs of branches i and j that share the node. Kirchhoff's current law states that

$$\sum_i \frac{1}{r_i}\,\frac{\partial \bar{G}_i(x,s)}{\partial x}\bigg|_{x\,=\,x_{\text{node}}} = 0, \tag{5.9}$$

where the sum is over the i branches adjacent to a node, and r_i is the longitudinal resistance per unit length of branch i.

At terminals, we will consider closed dendritic tips :

$$\left.\frac{\partial \bar{G}_i(x,s)}{\partial x}\right|_{x\,=\,x_{\mathrm{term}}} = 0. \tag{5.10}$$

5.2 Motifs

The motivation behind the use of motifs lies in the coefficients $\alpha(s)$ and $\beta(s)$ in (5.5). Obtaining expressions for these would enable us to compute the exact Laplace-domain solutions to the cable equation explicitly. Depending on the form of the solution used, there is one $(\alpha(s), \beta(s))$ pair, for each branch on the tree, and thus $2|\mathcal{E}|$ equations for these coefficients.

The branching nodes and terminal points on a tree are where the boundary conditions in Section 5.1.2 are applied, and thus, make a natural focus for the construction of the set of equations required to evaluate the coefficients in (5.5). Motifs are therefore centred around the nodes, allowing precomputed expressions to be used to construct a system of equations for these coefficients, which may then be solved in order to evaluate the solutions to the cable equation itself.

5.2.1 The Motif Concept

Motifs are *improper* directed graphical substructures (but not valid subgraphs), consisting of only one node and up to three branches adjacent to the node. The *propriety* of a graphical structure designates whether its edges are defined as existing between two nodes. One node is insufficient to have an edge; *proper* edges are defined as a relation, or link, between *two* nodes, in the traditional graph-theoretic sense (see Section 4.1). Motifs, structures with only one node, are therefore improper structures, simply because they contain free-hanging edges with no nodes on the other end. We will continue to refer to these improper edges as simply edges or branches, and each will be contextually-associated with a given proper edge on the tree; each edge on a motif will therefore be indexed, and will relate to a given dendritic branch.

There is a non-contextual relation between the node in the motif and the nodes on the tree. A node $v \in \mathcal{V}$ at the heart of a motif refers always to the same node v on the tree, and only to that node. In this way, we can associate each node on the tree with one motif in a surjective fashion. The improper edges on motifs, however, are not uniquely related to a branch on the tree. Instead, the association between motif edges and tree branches is contextual : the formation of a simple, order-2 binary tree (a four-node tree consisting of a central node with three proper edges radiating from it), we require four motifs, as each motif only has one node. Three of the four motifs will consist of one edge adjacent to the node, and the last motif will have three edges adjacent to the node. By "overlapping" the improper edges of the three one-edge motifs onto the three edges of the three-node motif, we are able to form a proper graphical structure with four nodes and three edges. This is illustrated in Figure 5.1. In this example, each node is associated with a motif from the set of motifs, $\mathcal{M}$, with certain motifs (specifically, the one with a single edge) being associated with several nodes on the tree (specifically, the terminals).

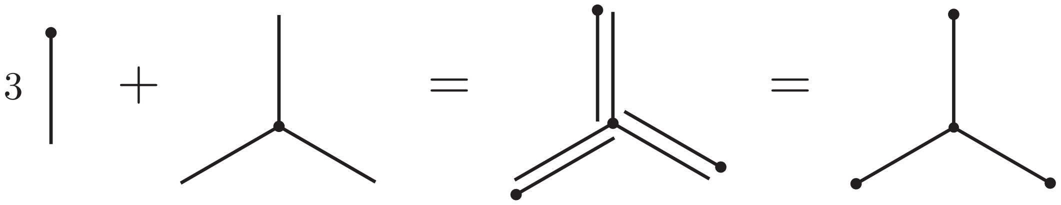


Figure 5.1: The construction of an order-2 binary tree from three motifs with one edge, and one three-edge motif.

When combined, motifs can be used as building blocks to form any tree we wish, up to a bounded degree $d = 3$, a biologically-relevant constraint. This constraint is merely imposed for simplicity; the ideas can be generalised to any bounded degree, although the number of calculations to be performed before a motif method can be used increases rapidly with d : a larger degree bound on the tree implies that the set of motifs required to construct the tree is also larger. The set of motifs, $\mathcal{M}$, is therefore a finite, countable set, whose cardinality depends non-trivially on the maximum degree on the tree.

5.2.2 Edge Orientation

Whilst dendritic morphologies are undirected graphs, imposing a direction on their edges can be algorithmically beneficial. In Section 4.3.2, where each branch was assigned a direction according to a given rule, then some level of ambiguity was removed from the algorithm - the edge directionality imposed by the Improved Four Classes algorithm stopped the production of duplicate trips.

Here, the directing of branches on tree imposes a choice of coordinates, such that $x = 0$ at the tail of a directed edge, and $x = L_i$ at the head of edge i. With regards to motifs, the directing of branches on a tree is a mechanism for ensuring that boundary conditions are applied only in the correct places and that they are not applied multiple times, because the coordinate system is now unambiguous.

Directed edges, however, increase the total number of possible motifs that are needed to construct the tree. Instead of having only one motif with a single edge, we now have two : one with the motif's node at the tail of the edge, and one with the node at its head. We must therefore be careful in our choice of rule for allocating each edge with a direction, in order to keep $|\mathcal{M}|$ to a minimum.

One rule which optimises over this constraint is to select any terminal and denote it the root node $v_R \in \mathcal{V}$, and to have all edges pointing away from the root node. This ensures that all nodes except the root v_R have one edge head, and anywhere between zero and two edge tails, adjacent to them. This is equivalent to saying that each node (again, except the root) houses the distal coordinate $x = L_i$ of one and only one edge, and the proximal coordinates $x = 0$ for between zero and two edges.

5.2.3 Nomenclature : The Set of Motifs

Motifs are given a name in the form X_Y, where $X = d(v)$ is the degree of the node $v \in \mathcal{V}$ in the motif, and Y is an indicator of the edge on which current is injected, if any on this motif. $X \in \{1, 2, 3\}$ by the degree constraint imposed in Section 5.2.1, and $Y \in \{0, i, j, k\}$ where 0 implies that no edge on this motif is being stimulated by an applied current I_{app}, and the indices i, j, k refer to a specific edge on the motif where current is injected. The exception to this is the root node v_R, which has its own motif, denoted 1_R. Figure 5.2 uses this notation to illustrate the complete set of motifs required for the construction of an arbitrary dendritic tree, given the upper bound on branching degree and the rule for directing branches.

The Motif method constructs a system of equations for the coefficients $\alpha(s)$ and $\beta(s)$ in (5.5) by precalculating the coefficients of the edges whose tails are at the node, from those whose head is at the node. Because each motif's node is the head of only one edge, each motif expresses how the daughter branches' coefficients, whose tails are at the node, can be expressed as a function of the mother edge's coefficients on the motif. The Forward Motif method simply constructs a linear system of these equations, which can then be inverted to solve for the coefficients.

5.3 Forward Motif Method

In this section, we will derive the expressions relating the coefficients on daughter branches to those on the mother branch, used in the Forward Motif method to obtain a system of equations for the coefficients $\alpha(s)$ and $\beta(s)$. First, motifs for branching nodes will be derived, followed by terminal motifs.

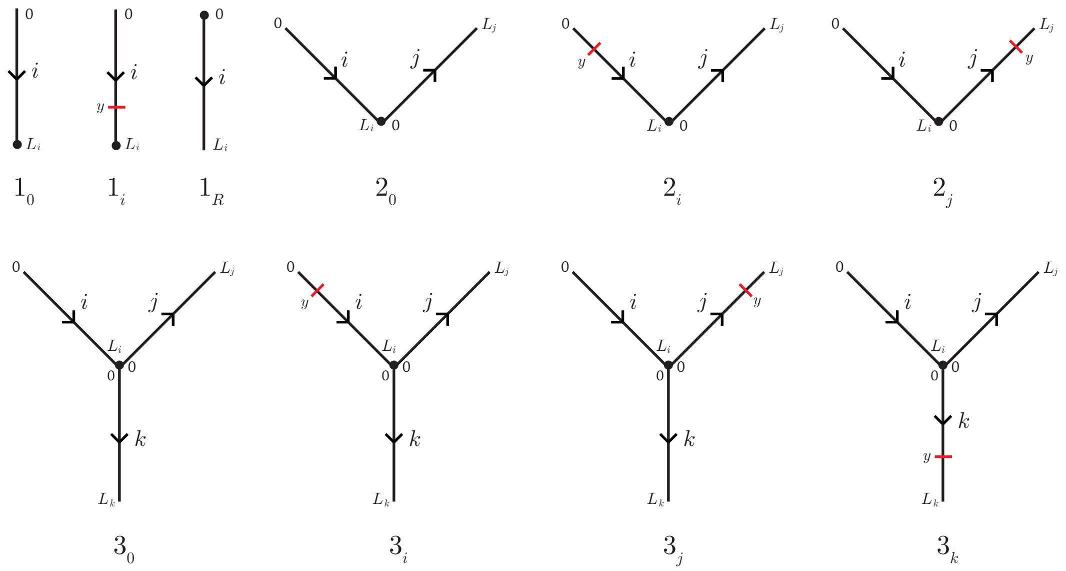


Figure 5.2: The complete set of motifs required for the construction of an arbitrary dendritic tree.

5.3.1 Coefficient Expressions

2_0 : Degree-2 Motif with No Injection

The 2_0 motif, as in Figure 5.2, is a two-branch system with mother edge $i \in \mathcal{E}$ pointing into the node, and daughter edge $j \in \mathcal{E}$ pointing out of it. With no injections of current on either i or j, this system is described by the set of equations,

$$\begin{aligned} G_i(x,s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x}, \\ G_j(x,s) &= \alpha_j(s)\,\mathrm{e}^{-\gamma_j(s)\,x} + \beta_j(s)\,\mathrm{e}^{\gamma_j(s)\,x}. \end{aligned} \tag{5.11}$$

At the node, the continuity of potential condition (5.8) implies that

$$G_i(L_i,s) = G_j(0,s), \tag{5.12}$$

and Kirchhoff's law (5.9) states that

$$\frac{1}{r_i}\frac{\partial G_i(x,s)}{\partial x}\bigg|_{x=L_i} = \frac{1}{r_j}\frac{\partial G_j(x,s)}{\partial x}\bigg|_{x=0}. \tag{5.13}$$

From (5.12), we find that

$$\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i} = \alpha_j(s) + \beta_j(s), \tag{5.14}$$

and (5.13) gives us

$$\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\Big(\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}-\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\Big)=\beta_j(s)-\alpha_j(s). \tag{5.15}$$

From equations (5.14) and (5.15), we find

$$\begin{aligned}\alpha_j(s)&=\alpha_i(s)\,\frac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)+\beta_i(s)\,\frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right),\\ \beta_j(s)&=\alpha_i(s)\,\frac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)+\beta_i(s)\,\frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right).\end{aligned} \tag{5.16}$$

We have successfully expressed the coefficients $\alpha(s)$ and $\beta(s)$ for the daughter branch j in terms of those of the mother branch i. Using the same approach, we can derive functions for coefficients of branches on the other motifs in the set of motifs. Once expressions are found for every required motif, they can be combined in a linear system of equations and solved by matrix inversion.

2_i : Degree-2 Motif with Injection on the Mother Branch

This motif is based on the same graphical structure as in 2_0, except that the mother branch i is now the subject to a current injection. The equations describing this motif are therefore

$$\begin{aligned}G_i(x,s)&=\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x}+\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x}-I_i^s(x,y,s),\\ G_j(x,s)&=\alpha_j(s)\,\mathrm{e}^{-\gamma_j(s)\,x}+\beta_j(s)\,\mathrm{e}^{\gamma_j(s)\,x}.\end{aligned} \tag{5.17}$$

Substituting in the boundary conditions (5.8) and (5.9) gives

$$\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}+\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}-I_i^s(L_i,y,s)=\alpha_j(s)+\beta_j(s), \tag{5.18}$$

$$\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\Big(\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}-\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}-I_i^c(L_i,y,s)\Big)=\beta_j(s)-\alpha_j(s), \tag{5.19}$$

respectively. From (5.18) and (5.19), we find that

$$
\begin{aligned}
\alpha_j(s) &= \alpha_i(s)\,\frac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)+\beta_i(s)\,\frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)\\
&\quad-\frac{1}{2}\left(I_i^s(L_i,y,s)-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\,I_i^c(L_i,y,s)\right),\\
\beta_j(s) &= \alpha_i(s)\,\frac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)+\beta_i(s)\,\frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)\\
&\quad-\frac{1}{2}\left(I_i^s(L_i,y,s)+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\,I_i^c(L_i,y,s)\right),
\end{aligned}
\tag{5.20}
$$

where I_j^c is defined as in (5.7).

2_j : Degree-2 Motif with Injection on the Daughter Branch

If the injected current is applied to branch j, then (5.5) fully describes this motif. Applying the continuity condition (5.8) gives us

$$
\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}+\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}=\alpha_j(s)+\beta_j(s), \tag{5.21}
$$

exactly as in (5.14), while the Kirchhoff boundary condition gives

$$
\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\left(\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}-\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\right)=\beta_j(s)-\alpha_j(s), \tag{5.22}
$$

as in (5.15). Thus, the 2_j motif provides the same expressions for the daughter branch's coefficients as the 2_0 motif :

$$
\begin{aligned}
\alpha_j(s) &= \alpha_i(s)\,\frac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)+\beta_i(s)\,\frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right),\\
\beta_j(s) &= \alpha_i(s)\,\frac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1-\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right)+\beta_i(s)\,\frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1+\frac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right).
\end{aligned}
\tag{5.23}
$$

3_0 : Degree-3 Motif with No Injection

A node at which a mother edge i enters and two daughter edges j and k leave, forms the 3_0 motif if none of the edges are subject to an injection of current. As we can see from (5.5), the equations that describe this system are

$$\begin{aligned}
G_i(x,s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x},\\
G_j(x,s) &= \alpha_j(s)\,\mathrm{e}^{-\gamma_j(s)\,x} + \beta_j(s)\,\mathrm{e}^{\gamma_j(s)\,x},\\
G_k(x,s) &= \alpha_k(s)\,\mathrm{e}^{-\gamma_k(s)\,x} + \beta_k(s)\,\mathrm{e}^{\gamma_k(s)\,x},
\end{aligned} \tag{5.24}$$

with boundary conditions

$$G_i(L_i,s) = G_j(0,s) = G_k(0,s), \tag{5.25}$$

and

$$\frac{1}{r_i}\frac{\partial G_i(x,s)}{\partial x}\bigg|_{x=L_i} = \frac{1}{r_j}\frac{\partial G_j(x,s)}{\partial x}\bigg|_{x=0} + \frac{1}{r_k}\frac{\partial G_k(x,s)}{\partial x}\bigg|_{x=0}. \tag{5.26}$$

From (5.25), we get

$$\alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i} = \alpha_j(s) + \beta_j(s) = \alpha_k(s) + \beta_k(s), \tag{5.27}$$

and (5.26) gives us

$$\frac{\gamma_i(s)}{r_i}\Big(\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i} - \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\Big) = \frac{\gamma_j(s)}{r_j}\Big(\beta_j(s) - \alpha_j(s)\Big) + \frac{\gamma_k(s)}{r_k}\Big(\beta_k(s) - \alpha_k(s)\Big). \tag{5.28}$$

Substituting (5.27) into (5.28), we obtain

$$\begin{aligned}
\frac{2\gamma_j}{r_j}\,\alpha_j(s) + \frac{2\gamma_k(s)}{r_k}\,\alpha_k(s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\left(\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right)\\
&\quad + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}\left(-\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right)\\
\frac{2\gamma_j}{r_j}\,\beta_j(s) + \frac{2\gamma_k(s)}{r_k}\,\beta_k(s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\left(-\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right)\\
&\quad + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}\left(\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right).
\end{aligned} \tag{5.29}$$

3_i : Degree-3 Motif with Injection on the Mother Branch

An injection on the mother branch on a 3_0 motif turns it into a 3_i motif. This system is described by the equations

$$\begin{aligned} G_i(x,s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x} - I_i^s(x,y,s), \\ G_j(x,s) &= \alpha_j(s)\,\mathrm{e}^{-\gamma_j(s)\,x} + \beta_j(s)\,\mathrm{e}^{\gamma_j(s)\,x}, \\ G_k(x,s) &= \alpha_k(s)\,\mathrm{e}^{-\gamma_k(s)\,x} + \beta_k(s)\,\mathrm{e}^{\gamma_k(s)\,x}, \end{aligned} \tag{5.30}$$

with the same boundary condition as for the 3_0 motif, (5.25) and (5.26). The first of these gives us

$$\begin{aligned} \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i} - I_i^s(L_i,y,s) &= \alpha_j(s) + \beta_j(s) \\ &= \alpha_k(s) + \beta_k(s), \end{aligned} \tag{5.31}$$

and the second provides us with

$$\begin{aligned} \frac{\gamma_i(s)}{r_i}\Big(\beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i} - \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i} - I_i^c(L_i,y,s)\Big) &= \frac{\gamma_j(s)}{r_j}\Big(\beta_j(s) - \alpha_j(s)\Big) \\ &\quad + \frac{\gamma_k(s)}{r_k}\Big(\beta_k(s) - \alpha_k(s)\Big). \end{aligned} \tag{5.32}$$

Proceeding as for the 3_0 motif, we substitute (5.31) into (5.32) and obtain

$$\begin{aligned} \frac{2\gamma_j}{r_j}\,\alpha_j(s) + \frac{2\gamma_k(s)}{r_k}\,\alpha_k(s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\left(\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right) \\ &\quad + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}\left(-\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right) \\ &\quad - \left(\frac{\gamma_i(s)}{r_i}\,I_i^c(L_i,y,s) - \Big(\frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\Big)I_i^s(L_i,y,s)\right). \\ \frac{2\gamma_j}{r_j}\,\beta_j(s) + \frac{2\gamma_k(s)}{r_k}\,\beta_k(s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\left(-\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right) \\ &\quad + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}\left(\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right) \\ &\quad - \left(\frac{\gamma_i(s)}{r_i}\,I_i^c(L_i,y,s) + \Big(\frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\Big)I_i^s(L_i,y,s)\right). \end{aligned} \tag{5.33}$$

$3_j, 3_k$: Degree-3 Motifs with Injection on a Daughter Branch

For the same reason as the 2_j motif returns coefficient expressions exactly equal to those provided by 2_0, the motifs 3_j and 3_k give the same coefficient expressions as the motif 3_0. Subtituting the system equations into boundary conditions (5.8) and (5.9) implies that the injection terms I_j^s or I_k^s and their derivatives I_j^c and I_k^c are evaluated at $x = 0$, and according to their definitions in (5.6) and (5.7), equate to zero. Thus, for 3_j and 3_k motifs, we have

$$\begin{aligned}\frac{2\gamma_j}{r_j}\,\alpha_j(s) + \frac{2\gamma_k(s)}{r_k}\,\alpha_k(s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\left(\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right)\\ &\quad + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}\left(-\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right)\\ \frac{2\gamma_j}{r_j}\,\beta_j(s) + \frac{2\gamma_k(s)}{r_k}\,\beta_k(s) &= \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,L_i}\left(-\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right)\\ &\quad + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,L_i}\left(\frac{\gamma_i(s)}{r_i} + \frac{\gamma_j(s)}{r_j} + \frac{\gamma_k(s)}{r_k}\right).\end{aligned} \tag{5.34}$$

1_0 : Terminal Motif with No Injection

The 1_0 motif has a mother edge entering a node, and no outgoing daughter edges. It applies at all terminals with no injections, aside from at the root node v_R. The equation that describes it is

$$G_i(x,s) = \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x}, \tag{5.35}$$

with the closed tip terminal condition (5.10) applying at the node. With only one branch on this motif, we cannot obtain an expression relating coefficients between different branches. However, we can find a relation between the two coefficients on the same branch :

$$\alpha_i(s) = \beta_i(s)\,\mathrm{e}^{2\,\gamma_i(s)\,L_i}. \tag{5.36}$$

1_1 : Terminal Motif with Injection

The equation describing an injection on a terminal branch is

$$G_i(x,s) = \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x} - I_i^s(x,y,s). \tag{5.37}$$

The terminal condition (5.10) can be applied to give

$$\alpha_i(s) = \beta_i(s)\,\mathrm{e}^{2\gamma_i(s)\,L_i} - \frac{\mathrm{e}^{\gamma_i(s)\,L_i}}{\gamma_i}\,I_i^c(L_i, y, s). \tag{5.38}$$

1_R : Root Motif

The 1_R motif is only ever applied once per tree : at the root. Here, a single edge leaves the root node at $x = 0$, and hence, is described by the following equation, regardless of whether the branch houses a current injection :

$$G_i(x, s) = \alpha_i(s)\,\mathrm{e}^{-\gamma_i(s)\,x} + \beta_i(s)\,\mathrm{e}^{\gamma_i(s)\,x}. \tag{5.39}$$

The terminal condition (5.10) means that

$$\alpha_i(s) = \beta_i(s). \tag{5.40}$$

5.3.2 Motif Matrix Rows

Constructing a linear system from these equations can be made trivial by rewriting them as matrix equations :

$$MC = I, \tag{5.41}$$

where the coefficient expressions are written in the forms of rows of a matrix, M, and I is a column vector of the inhomogeneous terms found in deriving the motif expressions. C, therefore, is a column vector of the coefficients themselves. This is the final step of the precalculation for the Forward Motif method, and serves as a convenient summary of the coefficient expressions.

Motifs $2_0, 2_j$

$$\begin{aligned} MC &= \begin{pmatrix} \dfrac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1 + \dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & \dfrac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1 - \dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & -1 & 0 \\ \dfrac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1 - \dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & \dfrac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1 + \dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \\ \alpha_j(s) \\ \beta_j(s) \end{pmatrix}, \\ I &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{5.42}$$

Motif 2_i

$$MC = \begin{pmatrix} \dfrac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1+\dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & \dfrac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1-\dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & -1 & 0 \\ \dfrac{\mathrm{e}^{-\gamma_i(s)\,L_i}}{2}\left(1-\dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & \dfrac{\mathrm{e}^{\gamma_i(s)\,L_i}}{2}\left(1+\dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\right) & 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \\ \alpha_j(s) \\ \beta_j(s) \end{pmatrix},$$

$$I = \frac{1}{2}\begin{pmatrix} I_i^s(L_i,y,s) - \dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\,I_i^c(L_i,y,s) \\ I_i^s(L_i,y,s) + \dfrac{r_j\,\gamma_i(s)}{r_i\,\gamma_j(s)}\,I_i^c(L_i,y,s) \end{pmatrix}. \tag{5.43}$$

Motifs $3_0, 3_j, 3_k$

$$MC = \begin{pmatrix} \mathrm{e}^{-\gamma_i(s)\,L_i} & \mathrm{e}^{\gamma_i(s)\,L_i} & -1 & -1 & 0 & 0 \\ \mathrm{e}^{-\gamma_i(s)\,L_i} & \mathrm{e}^{\gamma_i(s)\,L_i} & 0 & 0 & -1 & -1 \\ \mathrm{e}^{-\gamma_i(s)\,L_i}\,z_+ & \mathrm{e}^{\gamma_i(s)\,L_i}\,z_- & -2\dfrac{\gamma_j(s)}{r_j} & 0 & -2\dfrac{\gamma_k(s)}{r_k} & 0 \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \\ \alpha_j(s) \\ \beta_j(s) \\ \alpha_k(s) \\ \beta_k(s) \end{pmatrix},$$

$$I = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{5.44}$$

where $z_+ = \left(\dfrac{\gamma_i(s)}{r_i} + \dfrac{\gamma_j(s)}{r_j} + \dfrac{\gamma_k(s)}{r_k}\right)$ and $z_- = \left(-\dfrac{\gamma_i(s)}{r_i} + \dfrac{\gamma_j(s)}{r_j} + \dfrac{\gamma_k(s)}{r_k}\right)$.

Motif 3_i

$$MC = \begin{pmatrix} \mathrm{e}^{-\gamma_i(s)\,L_i} & \mathrm{e}^{\gamma_i(s)\,L_i} & -1 & -1 & 0 & 0 \\ \mathrm{e}^{-\gamma_i(s)\,L_i} & \mathrm{e}^{\gamma_i(s)\,L_i} & 0 & 0 & -1 & -1 \\ \mathrm{e}^{-\gamma_i(s)\,L_i}\,z_+ & \mathrm{e}^{\gamma_i(s)\,L_i}\,z_- & -2\dfrac{\gamma_j(s)}{r_j} & 0 & -2\dfrac{\gamma_k(s)}{r_k} & 0 \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \\ \alpha_j(s) \\ \beta_j(s) \\ \alpha_k(s) \\ \beta_k(s) \end{pmatrix},$$

$$I = \begin{pmatrix} I_i^s(L_i, y, s) \\ I_i^s(L_i, y, s) \\ \left(\dfrac{\gamma_j(s)}{r_j} + \dfrac{\gamma_k(s)}{r_k}\right) I_i^s(L_i, y, s) - \dfrac{\gamma_i(s)}{r_i}\, I_i^c(L_i, y, s) \end{pmatrix}, \tag{5.45}$$

where $z_+ = \left(\dfrac{\gamma_i(s)}{r_i} + \dfrac{\gamma_j(s)}{r_j} + \dfrac{\gamma_k(s)}{r_k}\right)$ and $z_- = \left(-\dfrac{\gamma_i(s)}{r_i} + \dfrac{\gamma_j(s)}{r_j} + \dfrac{\gamma_k(s)}{r_k}\right)$.

Motif 1_0

$$\begin{aligned} MC &= \begin{pmatrix} -1 & \mathrm{e}^{2\gamma_i(s)\,L_i} \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \end{pmatrix}, \\ I &= \begin{pmatrix} 0 \end{pmatrix}. \end{aligned} \tag{5.46}$$

Motif 1_1

$$\begin{aligned} MC &= \begin{pmatrix} -1 & \mathrm{e}^{2\gamma_i(s)\,L_i} \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \end{pmatrix}, \\ I &= \left(\tfrac{\mathrm{e}^{\gamma_i(s)\,L_i}}{\gamma_i}\, I_i^c(L_i, y, s) \right). \end{aligned} \tag{5.47}$$

Motif 1_R

$$\begin{aligned} MC &= \begin{pmatrix} -1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_i(s) \\ \beta_i(s) \end{pmatrix}, \\ I &= \begin{pmatrix} 0 \end{pmatrix}. \end{aligned} \tag{5.48}$$

5.3.3 Constructing a Solution

At this point, construction of a solution is trivial : we merely treat the matrices M as blocks, and insert them into a matrix of size $2|\mathcal{E}| \times 2|\mathcal{E}|$, and concatenate the vectors C and I into vectors of size $2|\mathcal{E}|$. Using the same matrix equation (5.41) for the new, larger matrix M and vectors C and I, we can obtain analytical expressions for the coefficients using

$$C = M^{-1}I. \tag{5.49}$$

As an example, let us consider the tree in Figure 5.3A, with injection on branch $(2, 4)$. We will construct the matrix M and the vectors C and I for this system by identifying the motifs in order of their node identifiers.

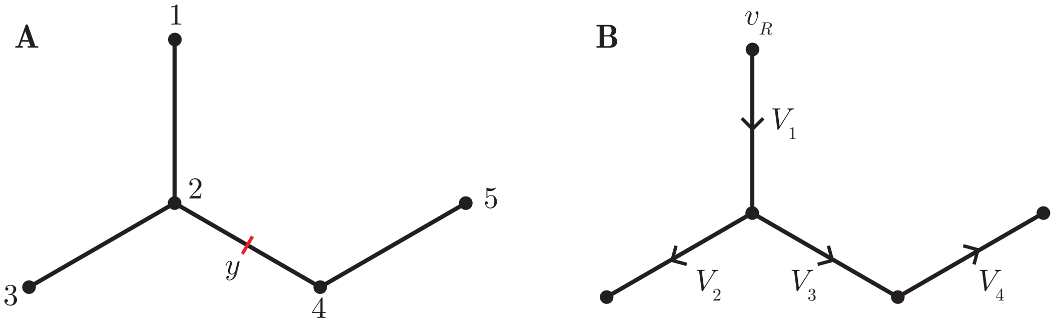


Figure 5.3: An example branching structure used in the construction of a solution using the Forward Motif method (**A**), with its equivalent structure (**B**) where a root node v_R has been nominated, edges oriented, and branches labelled V_1, V_2, V_3, V_4 according to their tail nodes.

We will nominate node 1 as the root. All edges therefore point away from this node, as in Figure 5.3B. Note that edges are identified by their tail node, such that the voltage along edge $(1, 2)$ is denoted V_1, and so on. Beginning with node 1, we select the 1_R motif, (5.48). Node 2 belongs to 3_j or 3_k equivalently, as an injection exists on one of its daughter branches. We therefore select (5.44). The next motif is 1_0 for the terminal node 3. Node 4 gives us 2_i, a degree-2 motif with injection on its mother branch. Finally, node 5 matches 1_0, another terminal with no injection.

We will first construct C, very simply as the column vector

$$C = \begin{pmatrix} \alpha_1 & \beta_1 & \alpha_2 & \beta_2 & \alpha_3 & \beta_3 & \alpha_4 & \beta_4 \end{pmatrix}^T , \tag{5.50}$$

where we ensure that the coefficients are in the same order as the nodes were evaluated. Then, the injection vector I is a vertical concatenation of the I vectors from the various motifs, giving us

$$I = \frac{1}{2}\left(0,\, 0,\, 0,\, 0,\, 0,\, \left(I_3^s(L_3, s) - \frac{r_4\,\gamma_3}{r_3\,\gamma_4}\, I_3^c(L_3, s)\right),\, \left(I_3^s(L_3, s) + \frac{r_4\,\gamma_3}{r_3\,\gamma_4}\, I_3^c(L_3, s)\right),\, 0\right)^T . \tag{5.51}$$

Finally, we must construct M by blocks. By inserting the correct coefficient expression matrices into the large matrix M, we find that

$$M = \left(\begin{array}{cccccccc}
\multicolumn{2}{c}{\boxed{M_A}} & 0 & 0 & 0 & 0 & 0 & 0 \\
\multicolumn{6}{c}{} & 0 & 0 \\
\multicolumn{6}{c}{\boxed{M_B}} & 0 & 0 \\
\multicolumn{6}{c}{} & 0 & 0 \\
0 & 0 & \multicolumn{2}{c}{\boxed{M_C}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \multicolumn{4}{c}{} \\
0 & 0 & 0 & 0 & \multicolumn{4}{c}{\boxed{M_D}} \\
0 & 0 & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\boxed{M_E}}
\end{array}\right), \tag{5.52}$$

where

$$M_A = \begin{pmatrix} -1 & 1 \end{pmatrix}, \tag{5.53}$$

$$M_B = \begin{pmatrix} \mathrm{e}^{-\gamma_1 L_1} & \mathrm{e}^{\gamma_1 L_1} & -1 & -1 & 0 & 0 \\ \mathrm{e}^{-\gamma_1 L_1} & \mathrm{e}^{\gamma_1 L_1} & 0 & 0 & 1- & -1 \\ \mathrm{e}^{-\gamma_1 L_1} z_+ & \mathrm{e}^{\gamma_1 L_1} z_- & -2\frac{\gamma_2}{r_2} & 0 & -2\frac{\gamma_3}{r_3} & 0 \end{pmatrix}, \tag{5.54}$$

$$M_C = \begin{pmatrix} -1 & \mathrm{e}^{2\gamma_2 L_2} \end{pmatrix}, \tag{5.55}$$

$$M_D = \begin{pmatrix} \frac{\mathrm{e}^{-\gamma_3 L_3}}{2}\left(1 + \frac{r_4\,\gamma_3}{r_3\,\gamma_4}\right) & \frac{\mathrm{e}^{\gamma_3 L_3}}{2}\left(1 - \frac{r_4\,\gamma_3}{r_3\,\gamma_4}\right) & -1 & 0 \\ \frac{\mathrm{e}^{-\gamma_3 L_3}}{2}\left(1 - \frac{r_4\,\gamma_3}{r_3\,\gamma_4}\right) & \frac{\mathrm{e}^{\gamma_3 L_3}}{2}\left(1 + \frac{r_4\,\gamma_3}{r_3\,\gamma_4}\right) & 0 & -1 \end{pmatrix}, \tag{5.56}$$

$$M_E = \begin{pmatrix} -1 & \mathrm{e}^{2\gamma_4 L_4} \end{pmatrix}. \tag{5.57}$$

The construction of the large matrix M from blocks of motif matrix rows, and of the injection and coefficient vectors C and I can be done using a symbolic computational package, such as Mathematica [2010] or the Symbolics Toolbox in Matlab [2012]. Once constructed, M can be inverted symbolically, and numerical values of s can be substituted into the matrix M^{-1} and the vector I. Each $M^{-1}(s)$ for different s is multiplied by $I(s)$, to obtain a frequency series $C(s)$, each element of which represents a coefficient $\alpha_i(s)$ or $\beta_i(s)$ for branch i.

Using these coefficients, we can evaluate (5.5) numerically and without approximation, obtaining a numerical solution for the Green's function $G_i(x, s)$ in the Laplace domain. The final step, then, is a numerical inverse Laplace transform, where the only errors for this method are introduced into the solution.

This algorithm's time complexity is fairly high, although it remains polynomial. Construction of the system of equations (by iterating over $|\mathcal{V}|$ nodes), takes $\mathcal{O}(|\mathcal{V}|)$ time. Once constructed, inverting the symbolic matrix costs $\mathcal{O}(|\mathcal{E}|^3)$ time.

In certain numerical inverse Laplace transform methods, such as Talbot's [1979] method, the number of times a Laplace-domain function must be sampled in order to invert it is equal to the number of time samples requested after inversion. Thus, substituting in different values of s into the dense symbolic matrix M^{-1} to obtain $G_i(x, t)$ up to some time T, sampled $T/\Delta t$ times in intervals of Δt milliseconds, takes $\mathcal{O}(|\mathcal{E}|^2\, T/\Delta t)$ time. This returns $T/\Delta t$ dense matrices, each of which must be multiplied by the vector I at a cost of $\mathcal{O}(|\mathcal{E}|^2)$. The total time complexity is therefore

$$\mathcal{O}(\underbrace{|\mathcal{V}|}_{\text{construction}} + \underbrace{|\mathcal{E}|^3}_{\text{inversion}} + T/\Delta t\,(\underbrace{|\mathcal{E}|^2}_{\text{substitution}} + \underbrace{|\mathcal{E}|^2}_{\text{multiplication}})), \tag{5.58}$$

should we invert the matrix M symbolically. However, it may be computationally faster to evaluate the matrix numerically for each s, and invert that many numerical matrices, rather than inverting the symbolic matrix once and evaluating the result for every s. Symbolic computation is extremely slow compared to numerical calculations, and there exist more optimal algorithms for numerical matrices. For example, if we first substitute the different s into M before its inversion, we take a cost of $\mathcal{O}(|\mathcal{E}|)$ instead of $\mathcal{O}(|\mathcal{E}|^2)$ due to M being sparse. Then, we must invert $T/\Delta t$ matrices instead of just one. However, solving $MC = I$ can be done using highly-efficient matrix decompositions instead of inversion, such as using the LU decomposition, which has the added advantage of preserving sparsity. This may well be more numerically stable than computing an inverse, and allows us to take advantage of the sparse matrix's structure, drastically reducing the computation time compared to the naive inversion and substitution of a symbolic matrix. We can calculate the time complexity, in this case, to be

$$\mathcal{O}(\underbrace{|\mathcal{V}|}_{\text{construction}} + T/\Delta t\,(\underbrace{|\mathcal{E}|}_{\text{substitution}} + \underbrace{|\mathcal{E}|^3}_{\text{LU}} + \underbrace{|\mathcal{E}|}_{\text{multiplication}})). \tag{5.59}$$

This method has the advantage of being incredibly simple to implement. Its major disadvantage, outside the fact that it requires a numerical inverse Laplace transform, is the symbolic inversion of a matrix, or the numerous inversions of numerical matrices : this is a limiting factor for both time and memory.

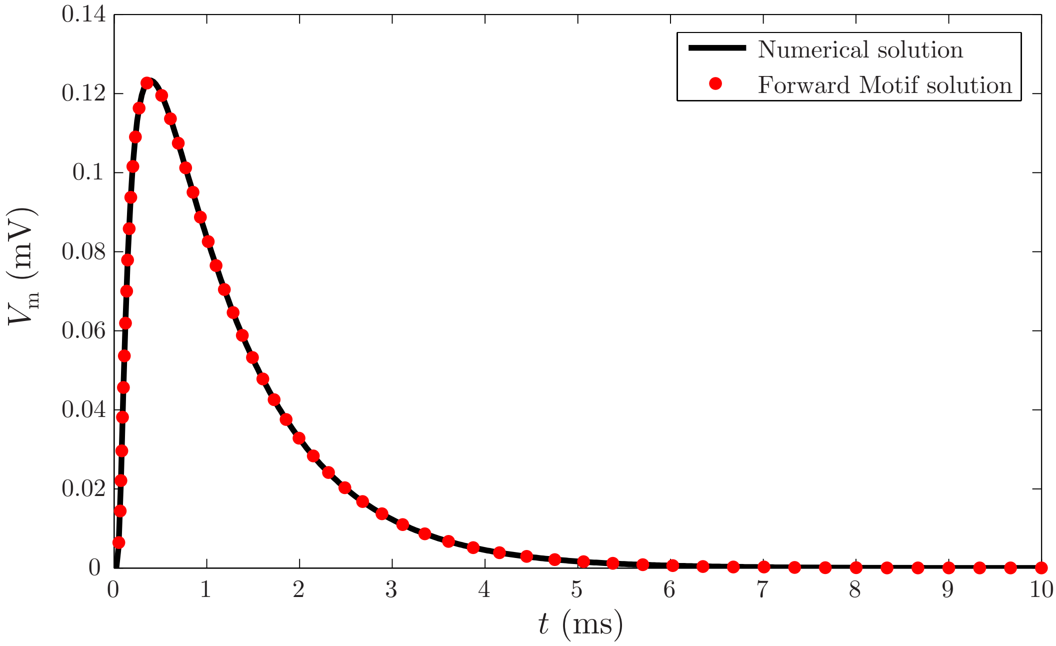


Figure 5.4: Transmembrane potential for the tree in Figure 5.3, as computed by the Forward Motif method, in black; the numerical solution computed using NEURON [Carnevale and Hines, 2006], in red.

With the Forward Motif method, the fastest and most accurate results for the construction and inversion of $G_i(x, s)$ to its time-domain representation, $G_i(x, t)$, were obtained using Talbot's [1979] method. Numerical instabilities were only observed should the point of injection y exist very close to the point of measurement, x. For the correct choice of Talbot [1979] method parameters, the solution demonstrated excellent agreement with the time-domain solution as calculated both numerically and using the Matrix method presented in Section 4.6, although this required the use of multiple precision libraries such as GMP [2012], which can lead to high computational costs on large trees. Figure 5.4 shows the transmembrane potential for the tree in Figure 5.3, computed using the Forward Motif method, against a solution computed numerically using NEURON [Carnevale and Hines, 2006]. The point of injection was placed halfway along the edge $(2, 4)$ and measurement was made halfway along edge $(1, 2)$. Results demonstrate a high accuracy, with an error of $\varepsilon = 1.1 \times 10^{-3}$ as computed in (4.19), a far smaller error than made with the Improved Four Classes, the Length Priority, or the Monte Carlo methods described in Chapter 4.

5.4 Conclusions

The Forward Motif method provides an exact analytical solution, with errors being introduced only in the numerical inversion. Its simplicity makes is easy to implement computationally, with significant computational gains to be made by parallelising the substitutions and matrix-vector products, despite the multiple precision arithmetic required for numerical stability in the Laplace transform inversion.

Chapter 6

Discussion

> “*Swiftly the brain becomes an enchanted loom, where millions of flashing shuttles weave a dissolving pattern – always a meaningful pattern – though never an abiding one.*”
>
> - Sherrington [1940]

6.1 Conclusions

In this thesis, we have studied the dynamics of the transmembrane potential on spatially-extended dendrites. Our focus was on developing and assessing novel algorithms, primarily built upon Abbott *et al.*'s [1991] path integral, for constructing solutions to the cable equation on arbitrary dendritic trees.

The importance of dendritic trees in neural systems was first brought to light by Wilfrid Rall, who demonstrated that the membrane properties of dendrites could be analysed rigorously [Rall, 1957, 1959, 1960]. Dendritic cable theory, Rall's [1959] groundbreaking description of the propagation of electrical signals in spatially-extended cells, provided a mathematical framework for the modelling of neurons with complex geometries, providing a means to assess the integrative function and signal processing properties of dendrites. Rall's work on cable theory methods and membrane properties, summarised in a book by Segev *et al.* [1995], inspired a large body of research in methods for constructing solutions to arbitrary dendritic trees. Approaches, approximate or exact, included Laplace transforms [Butz and Cowan, 1974; Horwitz, 1981; Koch and Poggio, 1985], reduction of the branching structure to simpler geometries [Poznanski, 1991; Whitehead and Rosenberg, 1993], infinite series solutions [Abbott *et al.*, 1991; Major *et al.*, 1993] and the construction of linear systems [Holmes, 1986]. Many methods have gone on to incorporate resonant or active dendrites [Koch, 1984; Coombes *et al.*, 2007], and we have seen how certain work seeks to assess how different dendritic trees integrate signal to and from the soma [Zador *et al.*, 1995; Mainen and Sejnowski, 1996], or how varying morphology may affect firing [Vetter *et al.*, 2001; Cook *et al.*, 2007]. An understanding of the relationship between dendritic structure and its impact on neuronal computation could be essential to an overarching theory of brain function, and lead to substantial advances in medicine.

6.1.1 Contributions

For such elaborate structures, however, a theoretical framework can be a far cry from a useable solution. Closed-form analytical solutions are typically infeasible to construct due to the sheer complexity of the dendritic trees, except on those with certain symmetries. As such, efficient algorithms must be developed to construct and evaluate a solution on arbitrary trees. Abbott *et al.*'s [1991] path integral for dendritic trees is a time-domain method for generating such solutions. Consisting of an infinite series, each term in the solution must be sampled from the tree. After proving that the dendritic path integral converges absolutely for an optimal ordering of the terms in the series solution, we successfully derived closed-form solutions for small, symmetrical trees, arranging the terms in the solution by the length of the trips on the tree. Our novel combinatorial counting scheme for symmetrical star graphs offers an optimal solution to the path integral, for a constrained subset of morphologies. These closed-form solutions do not explicitly construct trips, simply enumerating them, hence their high efficiency.

For arbitrary geometries, constructing trips individually is essential to sample from the complete tree. The trip length heuristic, used to determine how trips are sampled from the tree, is also at the core of Cao and Abbott's [1993] Four Classes algorithm. This algorithm provides a way to construct the series solution for any tree. After isolating some inefficiencies in this algorithm, we derived an equivalent formal grammar for sampling trips from a tree in increasing order of length, but sorted by their classes. This did not reduce the asymptotic time complexity of the algorithm in comparison to that provided by Cao and Abbott [1993], although its runtime is improved.

Then, based on Eppstein's [1999] efficient k shortest paths algorithm, we described a true Length Priority approach for constructing the series solution. Comparison of the convergence of the Improved Four Classes method and the Length Priority algorithm on real reconstructed morphologies demonstrated a strong dependence on the tree's geometry, and, more importantly, a slow and highly nonmonotonic convergence, indicating that length is a poor heuristic for term construction.

To address this, we developed a Monte Carlo approach by considering the Feynman-Kac relation between the cable theory equations and stochastic processes. This method was shown to have a far more stable and predictable (albeit slow) rate of convergence. The method is also trivial to implement, as it revolves around the sampling of random walkers on the nodes of the tree. This also makes the algorithm highly parallelisable on many-core systems.

A third approach, the Trip-Grouping Matrix method, was developed to take advantage of a way to compute the required properties of the trips (their lengths and coefficients) without explicitly constructing them. This method, applied to discretised dendritic structures, trades accuracy – discretisation of the tree into smaller segments reduces the error made but increases computation time and memory requirements – for extremely efficient computation costs. This allowed us to compute the error of convergence at many different times, to assess how simulation time affected the convergence of Abbott *et al.*'s [1991] path integral, which is known to be more accurate for short times. As expected, we demonstrated that a far greater number of trips is required to achieve the same level of accuracy for longer simulations.

These algorithms were used to evaluate some integrative properties of real dendritic morphologies. We showed how pyramidal cells propagate signals with less delay and attenuation than Purkinje cells, the latter of which are known experimentally to have a strong signal attenuation [Vetter *et al.*, 2001].

Exact solutions to cable problems can be found more readily in the Laplace domain. We derived a method for the construction of a linear system of equations from graphical motifs, which can be identified around each node on a dendritic tree. With a trivial computational implementation, the algorithm concatenates precomputed blocks into a matrix equation, which can then be solved to provide an exact solution in the Laplace domain. The frequency series were successfully inverted using Talbot's [1979] method for numerical Laplace transform inversion. Unlike the Four Classes, Length Priority, and Trip-Grouping algorithms, the Forward Motif method is parallelisable for performance on scientific computing systems.

The advantages to using analytical methods to construct the solution are numerous. The path integral provides a framework for computing the solution directly in the time domain which scales well with system size, allowing us to compute the transmembrane potential, or to calculate the Green's function, on large trees. This can be extended to networks of trees by allowing gap junctions at nodes. Length-priority approaches, despite potential convergence problems, can nonetheless provide solutions to high accuracy faster than numerical simulation. Stochastic sampling via the Monte Carlo method, which demonstrates predictable convergence, can be used on trees with pathological geometries. For large systems, a discretisation can be taken to allow the Trip-Grouping Matrix method to be used, yielding solutions in very short times and to very high accuracy. Finally, the Laplace-domain Motif method can be used to find exact analytical solutions to the cable equation on arbitrary geometries, with errors only being introduced during the system's numerical inverse transform.

6.2 Further Work

A primary aim for the future is the application of the methods described in this work to networks of neurons. Recent advances in imaging technology allow the three-dimensional reconstruction of not just individual neurons, but of small networks of connected neurons, such as the gap-junction-coupled GABAergic interneurons in the striatum or in the cerebral cortex [Fukuda, 2007, 2009], and, eventually, even the 302 chemically- and electrically-coupled neurons that comprise the full connectome of *C. elegans* [White *et al.*, 1986]. Calculation of the Green's function between the somas of such networks would enable us to rapidly compute the effect of current injections or spikes anywhere on the network. This function is interesting both for use in simulation, and in itself, as a network-level impulse response kernel.

The impulse response kernel provides us with insight on how signals diffuse along the network structure, and how much influence one neuron has over its proximal and distal neighbours. On the network-level, this gives us a quantitative description of signal integration, and allows a direct comparison between the electrotonic properties of different networks, via measures such as those of Zador *et al.* [1995], potentially providing insight into the "speed" or "range of impact" of signal processing on a network – measures which would take into account both somatic and dendritic time and space constants.

Using the methods presented in this thesis to compute impulse response functions has strong computational advantages. The Green's function allows the rapid calculation of the system's response to any type of stimulus, anywhere, by a simple convolution, whereas numerical methods require us to run a different simulation for every location of interest. Using Fukuda's [2009] networks of interneurons as an example, the Green's function $G_{ij}(x, y, t)$ would only need to be computed a small number of times, using the fact that a branch i need only be included if it forms an electrical synapse with another neuron; the Green's function on branches with no synapses are unnecessary for such a simulation. This makes a method such as the Matrix method from Section 4.6 ideal. Should the network consist of densely-connected neurons with relatively targetted branching (in comparison to space-filling branches as on Purkinje cells), then the Monte Carlo method, discussed in Section 4.5, could be used to high efficiency : each realisation of a random hop on nodes would provide a significant number of trips; in the best case, a random hopper on $k_{\max}$ nodes could contribute $k_{\max} - 1$ trips to $G_{ij}(x, y, t)$ for various i and j. On more sparsely-connected networks, application of Whitehead and Rosenberg's [1993] equivalent cable method could allow the reduction of branching to the point where the network becomes dense. Then, use of the reciprocal rule (3.116) halves the number of computations to be performed to obtain the impulse response function.

Naturally, such network-level computations requires us to be able to deal with synapses. The Forward Motif method developed in Section 5.3 is amenable to systems with electrical synapses by deriving motifs for gap-junction-coupled dendrites. This would enable the Green's function to be found on networks such as those in Fukuda [2007, 2009]. In addition, Laplace-domain methods such as the Forward Motif method support resonant dendrites as described by Koch and Poggio [1985], where dendritic filtration acts like a bandpass filter, rather than a lowpass filter.

Chemical synapses, being highly nonlinear, are more difficult, but can be simulated via a threshold process : should the transmembrane potential at the synapse reach the threshold at time $t = t_c$, a postsynaptic potential of the form $I_{\text{syn}}(t - t_c) \circledast G_{ij}(x, y, t)$ is initiated on the postsynaptic branch. Simulations can then be constructed of a sum of such Green's functions convolutions, delayed to account for the time taken for neurotransmitter to diffuse across the synaptic cleft, after the presynaptic terminal was brought to its firing threshold and began releasing neurotransmitter.

The final piece of the puzzle is the addition of active somas, such as Hodgkin and Huxley [1952] or Izhikevich [2003] model somas. By numerically simulating the ordinary differential equations driving soma dynamics, and using Green's function convolutions for whenever they spike, as with chemical synapses, we can do away with the spatial aspect of numerical integration, while retaining it within the system in the form of Green's functions. Then, for a network of N_n neurons connected across N_s chemical synapses, we would need to compute $N(N-1)/2$ Green's functions, with $N = N_\mathrm{n} + N_\mathrm{s}$. We then integrate N systems of ODEs numerically, adding the relevant Green's function to adjacent synapses should a soma fire, and to connected somas when a synapse initiates a postsynaptic potential. Compared with pure numerical simulation of the network, including the dendritic cables, a reduction of the network to its active ODEs and its Green's functions would allow significantly faster simulation of the network's dynamics. An ambitious aim, therefore, could be to develop a library, in the style of NeuroMorpho Ascoli *et al.* [2007], of Green's functions for various systems, allowing them to be precomputed once, and then made available for use in simulation.

Bibliography

L. F. Abbott. Simple diagrammatic rules for solving dendritic cable problems. *Physica A: Statistical Mechanics and its Applications*, 185 (1–4) : 343–356, **1992**.

L. F. Abbott and S. B. Nelson. Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3 : 1178–1183, **2000**.

L. F. Abbott, E. Farhi, and S. Gutmann. The path integral for dendritic trees. *Biological Cybernetics*, 66 (1) : 49–60, **1991**.

H. Agmon-Snir, C. E. Carr, and J. Rinzel. The role of dendrites in auditory coincidence detection. *Nature*, 393 (6682) : 268–272, **1998**.

F. Anselmi, C. Ventalon, A. Begue, D. Ogden, and V. Emiliani. Three-dimensional imaging and photostimulation by remote-focusing and holographic light patterning. *Proceedings of the National Academy of Sciences*, 108 (49) : 19504–19509, **2011**.

T. Araki and T. Otani. Response of single motoneurons to direct stimulation in toad's spinal cord. *Journal of Neurophysiology*, 18 (5) : 472–485, **1955**.

G. A. Ascoli, D. E. Donohue, and M. Halavi. NeuroMorpho.Org: a central resource for neuronal morphologies. *Journal of Neuroscience*, 27 (35) : 9247–9251, **2007**.

S. M. Baer and J. Rinzel. Propagation of dendritic spikes mediated by excitable spines: a continuum theory. *Journal of Neurophysiology*, 65 (4) : 874–890, **1991**.

S. Baigent, J. Stark, and A. Warner. Modelling the effect of gap junction nonlinearities in systems of coupled cells. *Journal of Theoretical Biology*, 186 (2) : 223–239, **1997**.

A. Bloomfield and F. Miller. A functional organization of ON and OFF pathways in rabbit retina. *Journal of Neuroscience*, 6 (1) : 1–13, **1986**.

J. M. Bower and D. Beeman. *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* Springer, **1998**.

W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems.* Wiley, **2012**.

E. G. Butz and J. D. Cowan. Transient potentials in dendritic systems of arbitrary geometry. *Biophysical journal*, 14 (9) : 661–689, **1974**.

R. Cachope, K. Mackie, A. Triller, J. O'Brien, and A. E. Pereda. Potentiation of electrical and chemical synaptic transmission mediated by endocannabinoids. *Neuron*, 56 (6) : 1034–1047, **2007**.

J. Campbell. *The Improbable Machine: What the Upheavals in Artificial Intelligence Research Reveal About How the Mind Really Works.* Simon and Schuster, **1989**.

B. J. Cao and L. F. Abbott. A new computational method for cable theory problems. *Biophysical Journal*, 64 (2) : 303–313, **1993**.

N.T. Carnevale and M.L. Hines. *The NEURON Book.* Cambridge University Press, **2006**.

Q. Caudron, S. R. Donnelly, S. P. C. Brand, and Y. Timofeeva. Computational convergence of the path integral for real dendritic morphologies. *Journal of Mathematical Neuroscience*, 2 (11), **2012**.

J. Clark and R. Plonsey. The extracellular potential field of the single active nerve fiber in a volume conductor. *Biophysical Journal*, 8 (7) : 842–864, **1968**.

K. D. Cole, J. V. Beck, A. Haji-Sheikh, and B. Litkouhi. *Methods for obtaining Green's functions.* Taylor and Francis, **2011**.

K. S. Cole. Dynamic electrical characteristics of the squid giant axon membrane. *Archives des Sciences Physiologiques*, 3 : 253–258, **1949**.

K. S. Cole. *Membranes, Ions and Impulses.* University of California Press, **1968**.

K. S. Cole and A. L. Hodgkin. Membrane and protoplasm resistance in the squid giant axon. *Journal of General Physiology*, 22 (5) : 671–687, **1939**.

E. P. Cook, A. C. Wilhelm, J. A. Guest, Y. Liang, N. Y. Masse, and C. M. Colbert. The neuronal transfer function: contributions from voltage- and time-dependent mechanisms. *Progress in Brain Research*, 165 : 1–12, **2007**.

J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19 (90) : 297–301, **1965**.

S. Coombes and P. C. Bressloff. Solitary waves in a model of dendritic cable with active spines. *SIAM Journal of Applied Mathematics*, 61 (2) : 432–453, **2000**.

S. Coombes and P. C. Bressloff. Saltatory waves in the spike-diffuse-spike model of active dendritic spines. *Physical Review Letters*, 91 (2) : 028102, **2003**.

S. Coombes, Y. Timofeeva, C.-M. Svensson, G. J. Lord, K. Josić, S. J. Cox, and C. M. Colbert. Branching dendrites with resonant membrane: a "sum-over-trips" approach. *Biological Cybernetics*, 97 (2) : 137–149, **2007**.

J. S. Coombs, J. C. Eccles, and P. Fatt. The inhibitory suppression of reflex discharges from motoneurones. *Journal of Physiology*, 130 (2) : 396–413, **1955**.

K. P. Cosgrove, C. M. Mazure, and J. K. Staley. Evolving knowledge of sex differences in brain structure, function, and chemistry. *Biological Psychiatry*, 62 (8) : 847–855, **2007**.

M. Cremer. Zum kernleiterproblem. *Zeitschrift für Biologie*, 37 : 550–553, **1899**.

F. Crick. Do dendritic spines twitch? *Trends in Neuroscience*, 5 : 44–46, **1982**.

H. Cuntz, J. Haag, and A. Borst. Neural image processing by dendritic networks. *Proceedings of the National Academy of Sciences*, 100 : 11082–11085, **2003**.

H. Cuntz, F. Forstner, J. Haag, and A. Borst. The morphological identity of insect dendrites. *PLoS Computational Biology*, 4 (12), **2008**.

H. J. Curtis and K. S. Cole. Transverse electric impedance of the squid giant axon. *Journal of General Physiology*, 21 (6) : 757–765, **1938**.

Y. Dan and M.-m. Poo. Spike timing-dependent plasticity of neural circuits. *Neuron*, 44 (1) : 23–30, **2004**.

L. Davis, Jr. and R. Lorente de Nó. Contribution to the mathematical theory of the electrotonus. *Studies from the Rockefeller Institute for Medical Research*, 131 : 442–496, **1947**.

F. R. de Hoog, J. H. Knight, and A. N. Stokes. An improved method for numerical inversion of Laplace transforms. *SIAM Journal on Scientific and Statistical Computing*, 3 (3) : 357–366, **1982**.

O. Deiters. *Untersuchungen über die Lamina spiralis membranacea*. Henry et Cohen, **1860**.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 : 269–271, **1959**.

D. Drachman. Do we have brain to spare ? *Neurology*, 64 (12) : 2004–2005, **2005**.

S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17 (3) : 395–412, **1969**.

R. S. Eisenberg and E. A. Johnson. Three dimensional electrical field problems in physiology. *Progress in Biophysics*, 20 : 1–65, **1970**.

Y. Elgersma and A. J. Silva. Molecular mechanisms of synaptic plasticity and memory. *Current Opinion in Neurobiology*, 9 (2) : 209–213, **1999**.

D. Eppstein. Finding the k Shortest Paths. *SIAM Journal on Computing*, 28 (2) : 652–673, **1999**.

P. Fatt. Sequence of events in synaptic activation of a motoneurone. *Journal of Neurophysiology*, 20 (1) : 61–80, **1957**.

E. S. Fortune and G. J. Rose. Short-term synaptic plasticity contributes to the temporal filtering of electrosensory information. *Journal of Neuroscience*, 20 (18) : 7122–7130, **2000**.

J. Fourier. *Théorie analytique de la chaleur*. Firmin Didot, Imprimeur du Roi, **1822**.

K. Frank and M. G. F. Fuortes. Stimulation of spinal motoneurones with intracellular electrodes. *Journal of Physiology*, 134 (2) : 451–470, **1956**.

T. Fukuda. Structural organization of the gap junction network in the cerebral cortex. *Neuroscientist*, 13 (3) : 199–207, **2007**.

T. Fukuda. Network architecture of gap junction-coupled neuronal linkage in the striatum. *Journal of Neuroscience*, 29 (4) : 1235–1243, **2009**.

D. P. Gaver. Observing stochastic processes, and approximate transform inversion. *Operations Research*, 14 (3) : 444–459, **1966**.

R. Gillette. On the significance of neuronal giantism in gastropods. *The Biological Bulletin*, 180 (2) : 234–240, **1991**.

D. L. Glanzman. Common mechanisms of synaptic plasticity: minireview in vertebrates and invertebrates. *Current Biology*, 20 (1) : R31–R36, **2010**.

GMP. *Version 5.1.0.* The GNU Multiple Precision Arithmetic Library. Torbjörn Granlund and the GMP development team, **2012**.

C. Golgi. Sulla struttura della sostanza grigia del cervello. *Gazetta Medica Italiana, Lombardia*, 6 : 244–246, **1873**.

J. Harris and Y. Timofeeva. Intercellular calcium waves in the fire-diffuse-fire framework: Greens function for gap-junctional coupling. *Physical Review E*, 82 (5), **2010**.

M. Häusser. Synaptic function: dendritic democracy. *Current Biology*, 11 (1) : R10–R12, **2001**.

S. Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in Neuroscience*, 3 : 1–11, **2009**.

L. Hermann. Beiträge zur physiologie und physik des nerven. *Pflüger's Archiv für die gesamte Physiologie des Menschen und der Tiere*, 109 (3) : 95–144, **1905**.

V. M. Ho, J.-A. Lee, and K. C. Martin. The cell biology of synaptic plasticity. *Science*, 334 (6056) : 623–628, **2011**.

A. L. Hodgkin. The membrane resistance of a non-medulated nerve fibre. *Journal of Physiology*, 106 (3) : 305–318, **1947**.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117 (4) : 500–544, **1952**.

A. L. Hodgkin and B. Katz. The effect of sodium ions on the electrical activity of the giant axon of the squid. *Journal of Physiology*, 108 (1) : 37–77, **1949**.

A. L. Hodgkin and W. A. H. Rushton. The electrical constants of a crustacean nerve fibre. *Proceedings of the Royal Society B: Biological Sciences*, 131 (873) : 444–479, **1946**.

A. L. Hodgkin, A. F. Huxley, and B. Katz. Measurement of current-voltage relations in the membrane of the giant axon of the *Loligo*. *Journal of Physiology*, 116 (4) : 424–448, **1952**.

R. Hoffman and R. R. Pavley. A method for the solution of the n^{th} best path problem. *Journal of the Association for Computing Machinery*, 6 (4) : 506–514, **1959**.

W. R. Holmes. A continuous cable method for determining the transient potential in passive dendritic trees of known geometry. *Biological Cybernetics*, 55 (2–3) : 115–124, **1986**.

G. Holt. *A Critical Reexamination of Some Assumptions and Implications of Cable Theory in Neurobiology.* PhD thesis, California Institute of Technology, **1998**.

J. L. Hoorweg. Über die elektrischen eigenschaften der nerven. *Pflüger's Archiv für die gesamte Physiologie des Menschen und der Tiere*, 71 (3–4) : 128–157, **1898**.

H. Horstmann, C. Korber, K. Satzler, D. Aydin, and T. Kuner. Serial section scanning electron microscopy ((SEM)-E-3) on silicon wafers for ultra-structural volume imaging of cells and tissues. *PLoS ONE*, 7 (4) : e35172, **2012**.

B. Horwitz. An analytic method for investigation transient potentials in branched neurons with branching dendritic trees. *Biophysical Journal*, 36 (1) : 155–192, **1981**.

E. H. Hu, F. Pan, Béla Völgyi, and S. A. Bloomfield. Light increases the gap junctional coupling of retinal ganglion cells. *Journal of Physiology*, 588 (21) : 4145–4163, **2010**.

E. M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14 (6) : 1569–1572, **2003**.

E. M. Izhikevich and G. M. Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 105 (9) : 3593–3598, **2008**.

J. J. B. Jack and S. Redman. The propagation of transient potentials in some linear cable structures. *Journal of Physiology*, 215 (2) : 283–320, **1971**.

V. M. Jiménez and A. Marzal. A Lazy Version of Eppsteins K Shortest Paths Algorithm. *Experimental and Efficient Algorithms - Lecture Notes in Computer Science*, 2647 : 179–191, **2003**.

D. Johnston and R. Narayanan. Active dendrites: colorful wings of the mysterious butterflies. *Trends in Neurosciences*, 31 (6) : 309–316, **2008**.

D. Johnston, J. C. Magee, C. M. Colbert, and B. R. Christie. Active properties of neuronal dendrites. *Annual Review of Neuroscience*, 19 : 165–186, **1996**.

B. Katz. The electrical properties of the muscle fibre membrane. *Proceedings of the Royal Society B: Biological Sciences*, 135 (881) : 506–534, **1948**.

P. W. Keeley, I. E. Whitney, M. A. Raven, and B. E. Reese. Dendritic spread and functional coverage of starburst amacrine cells. *Journal of Comparative Neurology*, 505 (5) : 539–546, **2005**.

C. Koch. Cable theory in neurons with active, linearized membranes. *Biological Cybernetics*, 50 (1) : 15–33, **1984**.

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

C. Koch and T. Poggio. A simple algorithm for solving the cable equation in dendritic trees of arbitrary geometry. *Journal of Neuroscience Methods*, 12 (4) : 303–315, **1985**.

J. Kozloski and J. Wagner. An ultrascalable solution to large-scale neural tissue simulation. *Frontiers in Neuroinformatics*, 5 (15), **2011**.

J. L. Krichmar, S. J. Nasuto, R. Scorcioni, S. D. Washington, and G. A. Ascoli. Effects of dendritic morphology on CA3 pyramidal cell electrophysiology : a simulation study. *Brain Research*, 941 : 11–28, **2002**.

K. F. H. Lee, C. Soares, and J.-C. Bé̈ïque. Examining form and function of dendritic spines. *Neural Plasticity*, 2012 : 704103, **2012**.

K. Lindsay. Analytical and numerical construction of equivalent cables. *Mathematical Biosciences*, 184 (2) : 137–164, **2003**.

K. A. Lindsay, J. M. Ogden, and J. R. Rosenberg. Equivalence transformations for dendritic Y-junctions: a new definition of dendritic sub-unit. *Mathematical Biosciences*, 170 (2) : 133–154, **2001**.

M. London and M. Häusser. Dendritic computation. *Annual Review of Neuroscience*, 28 : 503–532, **2005**.

J. C. Magee and E. P. Cook. Somatic EPSP amplitude is independent of synapse location in hippocampal pyramidal neurons. *Nature Neuroscience*, 3 (9) : 895–903, **2000**.

J. C. Magee and D. Johnston. A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons. *Science*, 275 (5297) : 209–213, **1997**.

Z. F. Mainen and T. J. Sejnowski. Influence of Dendritic Structure on Firing Pattern in Model Neocortical Neurons. *Nature Letters*, 382 (6589) : 363–366, **1996**.

G. Major, J. D. Evans, and J. J. B. Jack. Solutions for transients in arbitrarily branching cables: I. Voltage recording with a somatic shunt. *Biophysical Journal*, 65 (1) : 423–449, **1993**.

F. Mammano. *Cell-Cell Channels*, chapter Gap Junctions: Cell-Cell Channels in Animals. Springer-Verlag, **2006**.

H. Markram. The Blue Brain Project. *Nature Reviews Neuroscience*, 6 (2) : 153–160, **2006**.

H. Markram. The Human Brain Project, `www.humanbrainproject.eu`, **2012**.

G. Marmont. Studies on the axon membrane. *Journal of Cellular and Comparative Physiology*, 34 (3) : 351–382, **1949**.

Mathematica. *Version 8.0.* Wolfram Research, Inc., **2010**.

M. C. Matteucci. Sur le pouvoir électromoteur secondaire des nerfs, et son application à l'électrophysiologie. *Comptes Rendus de l'Académie des Sciences*, 56 : 231–235, **1863**.

Matlab. *Version 2012a.* MathWorks Inc., **2012**.

E. A. Nimchinsky, B. L. Sabatini, and K. Svoboda. Structure and function of dendritic spines. *Annual Review of Physiology*, 64 : 313–353, **2002**.

Z. Padamsey and A. Jeans. Imaging synaptic vesicles using VGLUT1-Venus knock-in mice: insights into the dynamic nature of intersynaptic vesicle exchange. *Journal of Neuroscience*, 32 (10) : 3284–3286, **2012**.

A. Parent and M. B. Carpenter. *Carpenter's Human Neuroanatomy.* Williams & Wilkins, **1995**.

W. F. Pickard. Electrotonus on a cell of finite dimensions. *Mathematical Biosciences*, 10 (3–4) : 201–213, **1971**.

R. Plonsey. Volume conductor fields of action currents. *Biophysical Journal*, 4 (4) : 317–328, **1964**.

A. Polsky, B. W. Mel, and J. Schiller. Computational subunits in thin dendrites of pyramidal cells. *Nature Neuroscience*, 7 (6) : 621–627, **2004**.

E. L. Post. Generalized Differentiation. *Transactions of the American Mathematical Society*, 32 (4) : 723–781, **1930**.

R. R. Poznanski. A generalized tapering equivalent cable model for dendritic neurons. *Bulletin of Mathematical Biology*, 53 (3) : 457–467, **1991**.

N. Qian and T. J. Sejnowski. An electro-diffusion model for computing membrane potentials and ionic concentrations in branching dendrites, spines and axons. *Biological Cybernetics*, 62 (1) : 1–15, **1989**.

N. Qian and T. J. Sejnowski. When is an inhibitory synapse effective? *Proceedings of the National Academy of Sciences*, 87 (20) : 8145–8149, **1990**.

T. Radman, R. L. Ramos, J. C. Brumberg, and M. Bikson. Role of Cortical Cell Type and Morphology in Sub- and Suprathreshold Uniform Electric Field Stimulation. *Brain Stimulation*, 2 (4) : 215–228, **2009**.

W. Rall. Membrane time constant of motoneurons. *Science*, 126 (3271) : 454, **1957**.

W. Rall. Branching dendritic trees and motoneuron membrane resistivity. *Experimental Neurology*, 1 (5) : 491–527, **1959**.

W. Rall. Membrane potential transients and membrane time constant of motoneurons. *Experimental Neurology*, 2 (5) : 503–532, **1960**.

W. Rall. Electrophysiology of a dendritic neuron model. *Biophysical Journal*, 2 (2) : 145–167, **1962**a.

W. Rall. Theory of physiological properties of dendrites. *Annals of the New York Academy of Sciences*, 96 (4) : 1071–1092, **1962**b.

W. Rall. Theoretical significance of dendritic trees for neuronal input-output relations. In R. F. Reiss, editor, *Neural Theory and Modeling*, pages 73–97. Stanford University Press, **1964**.

W. Rall. Distributions of potential in cylindrical coordinates and time constants for a membrane cylinder. *Biophysical Journal*, 9 (12) : 1509–1541, **1969**.

W. Rall. Core conductor theory and cable properties of neurons. In E. R. Kandel, editor, *Handbook of Physiology - The Nervous System (I)*, pages 39–97. Oxford University Press, **1977**.

S. Ramón y Cajal. *Textura del sistema nervioso del hombre y de los vertebrados.* Moya, **1899**.

C. Rashbass and W. A. H. Rushton. The relation of structure to the spread of excitation in the frogs sciatic trunk. *Journal of Physiology*, 110 (1–2) : 110–135, **1949**.

P. Rosenfalck. *Intra- and Extracellular Potential Fields of Active Nerve and Muscle Fibres.* PhD thesis, Copenhagen University, **1969**.

B. L. Sabatini, T. G. Oertner, and K. Svoboda. The life cycle of Ca^{2+} ions in dendritic spines. *Neuron*, 33 (3) : 439–452, **2002**.

D. Satoh, R. Suyama, K. Kimura, and T. Uemura. High-resolution in vivo imaging of regenerating dendrites of Drosophila sensory neurons during metamorphosis: local filopodial degeneration and heterotypic dendrite-dendrite contacts. *Genes to Cells*, 17 (12) : 939–951, **2012**.

T. Schwann. *Microscopic investigations on the accordance in the structure and growth of plants and animals.* Berlin, English translation by the Sydenham Society, **1839**.

A. C. Scott. Effect of the series inductance of a nerve axon upon its conduction velocity. *Mathematical Biosciences*, 11 (3–4) : 277–290, **1971**.

I. Segev, J. Rinzel, and G. M. Shepherd. *The Theoretical Foundation of Dendritic Function.* MIT Press, **1995**.

G. M. Shepherd, R. K. Brayton, J. P. Miller, I. Segev, J. Rinzel, and W. Rall. Signal enhancement in distal cortical dendrites by means of interactions between active dendritic spines. *Proceedings of the National Academy of Sciences*, 82 (7) : 2192–2195, **1985**.

C. Sherrington. *Man on his Nature.* Cambridge University Press, **1940**.

N. Spruston and D Johnston. Perforated patch-clamp analysis of the passive membrane properties of three classes of hippocampal neurons. *Journal of Neurophysiology*, 67 (3) : 508–529, **1992**.

N. Spruston, G. Stuart, and M. Häusser. *Dendrites.* Oxford University Press, **1999**.

G. Stuart and N. Spruston. Determinants of voltage attenuation in neocortical pyramidal neuron dendrites. *Journal of Neuroscience*, 18 : 3501–3510, **1998**.

G. Stuart, N. Spruston, B. Sakmann, and M. Häusser. Action potential initiation and backpropagation in neurons of the mammalian CNS. *Trends in Neuroscience*, 20 (3) : 125–131, **1997**.

G. J. Stuart and M. Häusser. Dendritic coincidence detection of EPSPs and action potentials. *Nature Neuroscience*, 4 (1) : 63–71, **2001**.

C.-M. Svensson. *Dynamics of Spatially Extended Dendrites.* PhD thesis, University of Nottingham, **2009**.

A. Talbot. The accurate numerical inversion of Laplace transforms. *IMA Journal of Applied Mathematics*, 23 (1) : 97–120, **1979**.

R. E. Taylor. *Physical Techniques in Biological Research*, chapter Cable Theory, pages 219–262. New York Academic, **1963**.

W. R. Taylor and R. G. Smith. The role of starburst amacrine cells in visual signal processing. *Visual Neuroscience*, 29 (1) : 73–81, **2012**.

M. Thallmair, G. A. S. Metz, W. J. Z'Graggen, O. Raineteau, G. L. Kartje, and M. E Schwab. Neurite growth inhibitors restrict plasticity and functional recovery following corticospinal tract lesions. *Nature Neuroscience*, 1 (4) : 124–131, **1998**.

W. Thomson. On the theory of the electric telegraph. *Mathematical and Physical Papers*, 7 : 382–399, **1854**.

Y. Timofeeva. *Oscillations and Waves in Single and Multi-cellular Systems with Free Calcium.* PhD thesis, University of Loughborough, **2003**.

Y. Timofeeva, G. J. Lord, and S. Coombes. Spatio-temporal filtering properties of a dendritic cable with active spines: a modeling study in the spike-diffuse-spike framework. *Journal of Computational Neuroscience*, 21 (3) : 293–306, **2006**.

Y. Timofeeva, S. J. Cox, S. Coombes, and K. Josić. Democratization in a passive dendritic tree: an analytical investigation. *Journal of Computational Neuroscience*, 25 (2) : 228–244, **2008**.

H. C. Tuckwell. *Introduction to Theoretical Neurobiology: Volume 1, Linear Cable Theory.* Cambridge University Press, **1988**.

B. Ulfhake and J. O. Kellerth. A quantitative light microscopic study of the dendrites of cat spinal alpha-motoneurons after intracellular staining with horseradish peroxidase. *Journal of Comparative Neurology*, 202 (4) : 571–583, **1981**.

E. I. Vaney, B. Sivyer, and W. R. Taylor. Direction selectivity in the retina: symmetry and asymmetry in structure and function. *Nature Reviews Neuroscience*, 13 (3) : 194–208, **2012**.

P. Vetter, A. Roth, and M. Häusser. Propagation of Action Potentials in Dendrites Depends on Dendritic Morphology. *Journal of Neurophysiology*, 85 (2) : 926–937, **2001**.

H. Weber. Über die stationären strömunger der elektricität in cylindern. *Journal für die Reine und Angewandte Mathematik*, 76 (1) : 1–20, **1873**.

W. T. Weeks. Numerical inversion of Laplace transforms using Laguerre functions. *Journal of the Association for Computing Machinery*, 13 (3) : 419–429, **1966**.

Wellcome Library, London. Wellcome Images, WWW.WELLCOMEIMAGES.ORG.

E. White, J. G. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode caenorhabditis elegans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 314 (1165) : 1–340, **1986**.

R. R. Whitehead and J. R. Rosenberg. On trees as equivalent cables. *Proceedings of the Royal Society B: Biological Sciences*, 252 (1334) : 103–108, **1993**.

J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17 (11) : 712–716, **1971**.

K. Yoshida, D. Watanabe, H. Ishikane, M. Tachibana, I. Pastan, and S. Nakanishi. A key role of starburst amacrine cells in originating retinal directional selectivity and optokinetic eye movement. *Neuron*, 30 (3) : 771–780, **2001**.

J. Z. Young. The structure of nerve fibres in cephalopods and crustacea. *Proceedings of the Royal Society B: Biological Sciences*, 121 (823) : 319–337, **1936**.

A. M. Zador, H. Agmon-Snir, and I. Segev. The morphoelectrotonic transform: a graphical approach to dendritic function. *Journal of Neuroscience*, 15 (3) : 1669–1682, **1995**.

R. S. Zucker and W. G. Regehr. Short-term synaptic plasticity. *Annual Review of Physiology*, 64 : 355–405, **2002**.