

Using the Green's function to simplify and understand dendrites

THÈSE N° 7869 (2017)

PRÉSENTÉE LE 21 SEPTEMBRE 2017
À LA FACULTÉ DES SCIENCES DE LA VIE
PROJET BLUEBRAIN
PROGRAMME DOCTORAL EN NEUROSCIENCES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Willem Anna Mark WYBO

acceptée sur proposition du jury:

Prof. W. Gerstner, président du jury
Prof. H. Markram, Dr M.-O. Gewaltig, directeurs de thèse
Prof. W. Senn, rapporteur
Dr A. Roth, rapporteur
Prof. K. Hess, rapporteuse



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2017

“No wonder kids grow up crazy. A cat’s cradle is nothing but a bunch of X’s between somebody’s hands, and little kids look and look and look at all those X’s. . .”

“And?”

“No damn cat, and no damn cradle.”

— Kurt Vonnegut

To my parents. . .

Acknowledgements

None of the results in this thesis would have been achieved without the support of key people. First, I would like to thank Dr. Benjamin Torben-Nielsen, who can not be listed as co-supervisor due to legal reasons but who deserves no less. He convinced me that the effort to simplify a detailed cortical network model in an informed way requires understanding and accurately reducing the dendritic tree, and pointed me in the direction of the Green's function as a tool to do so. During the past five years, he has kept me on track during my mathematical wanderings; I knew that if I could intuitively explain an idea to him, it had potential. I am also indebted to Dr. Marc-Oliver Gewaltig, who provided the freedom and ample advice to achieve my goals, and who was always supportive and patient, even when research avenues proved to be dead ends and many months of work had to be thrown in the bin. I am also grateful to Prof. Henry Markram, for whom the bigger picture is always more important than short term gain, for creating a stable environment where one can pursue long and profound projects. I would furthermore like to thank Prof. Idan Segev and Prof. Wulfram Gerstner. Discussing with them was always helpful, and residing in their labs during this thesis work was an enlightening and enriching experience.

When manuscripts had to be written, Dr. Luc Guyot proved to be a most careful and precise proofreader. His experience in mathematics also helped greatly in ordering mathematical ideas into theorems, lemmas and propositions. At the start of this thesis work, a computer, to me, was a magic box spitting out errors. Daniel Peppicelli, Csaba Erő, Werner Van Geit and many others helped change this. Completing a thesis is an endurance effort, and as such requires ample hydration. I look back fondly on the many great evenings I spent with many great people in Lausanne and Geneva.

Finally, I would like to thank my family and friends back in Ghent, Belgium. Taking off requires a smooth runway. My parents provided this runway, and gave me the means and encouragement to try to achieve whatever I wanted. Whenever I came back, they, as well as my friends, made it feel like I had never even left. For more I could not ask.

Lausanne, 07/05/2017

Willem Wybo

Abstract

Neurons are endowed with dendrites; specialized, tree-like structures that collect and transform inputs. These arborizations have been the subject of intense study in the past years, as they are believed to substantially enhance the computational repertoire of neurons. While it has long been known that dendrites are *not* iso-potential units, only in the last few decades it was shown experimentally that dendritic branches can transform local inputs in a non-linear fashion. This finding provides evidence for the *subunit hypothesis*, which states that within the dendritic tree, inputs arriving in one branch are transformed non-linearly and independently from what happens in other branches. Recent progress in experimental recording techniques shows that this highly localized dendritic integration contributes to shaping behaviour.

While it is generally accepted that the dendritic tree induces multiple subunits, many questions remain unanswered. For instance, it is not known how much separation there needs to be between different branches to be able to function as subunits. Consequently, there is no information on how many subunits can coexist along a dendritic arborization. It is also not known what the input-output relation of these subunits would be, or whether these subunits can be modified by input patterns. As a consequence, assessing the effects of dendrites on the workings of networks of neurons remains mere guesswork.

During this work, we choose a theory-driven approach to advance our knowledge about dendrites. Theory can help us understand dendrites by deriving accurate, but conceptually simple models of dendrites that still capture their main computational effects. The conceptually simple models can then be analyzed and fully understood, which in turn teaches us how actual dendrites function computationally. Such simple models typically require less computer operations to simulate than highly detailed dendrite models. Hence, these models increase the speed of network simulations that incorporate dendrites, so that studying the effects of dendrites on the workings of neural networks becomes easier.

The Green's function forms the basis for our theory driven approach. We first explored whether the Green's function could be used to reduce the cost of simulating dendrite models. One mathematically interesting finding in this regard is that the dendritic tree induces a special structure in the Green's function, so that the number of equations required to model the dendritic tree reduces drastically. Nevertheless, we were forced to conclude that reducing dendrites in this way does not yield new information about the subunit hypothesis.

We then focused our attention on another way of decomposing the Green's function. We found that the dendrite model obtained in this way reveals much information on the dendritic subunits. In particular, we found that the occurrence of subunits is well predicted by the ratio

Acknowledgements

of input over transfer impedance in dendrites. This allowed us to estimate the number of subunits that can coexist on dendritic trees. We also found that this ratio can be modified by other inputs, in particular shunting conductances, so that the number of subunits on a dendritic tree can be modified dynamically. We finally were able to show that, due to this dynamical increase of the number of subunits, individual branches that would otherwise respond to inputs as a single unit, could become sensitive to different stimulus features. We believe that this model can be implemented in such a way that it simulates dendrites in a highly efficient manner. Thus, after incorporation in standard neural network simulation software, it can substantially improve the accessibility of dendritic network simulations to modelers.

Key words: Dendrites, Subunits, Simplified models, Green's function

Résumé

Les neurones sont équipés de dendrites : des arborisations spécialisées qui reçoivent et transforment des signaux d'entrée. Ces arborisations ont été étudiées abondamment ces dernières années, car on les soupçonne d'étendre fortement les possibilités de calcul des neurones. Bien qu'il soit connu depuis longtemps que les dendrites ne sont pas des unités équipotentiels, c'est seulement dans les trois dernières décennies qu'il a été démontré expérimentalement que les branches dendritiques individuelles peuvent transformer des signaux d'entrées d'une façon non linéaire. Ce résultat constitue une preuve de *l'hypothèse des sous-unités computationnelles*, qui dit que dans les dendrites, les signaux d'entrées qui arrivent sur une branche sont transformés non linéairement et indépendamment de ce qui se passe dans les autres branches. Des progrès récents des techniques d'enregistrement montrent que cette intégration dendritique localisée contribue à produire certains comportements chez les rongeurs.

Tandis qu'il est généralement accepté qu'un arbre dendritique induit plusieurs de ces sous-unités, beaucoup de questions restent sans réponse. Par exemple, on ignore à quel point deux branches doivent être séparées pour qu'elles puissent fonctionner comme des sous-unités indépendantes. De ce fait, on ignore combien de sous-unités peuvent coexister sur le même arbre dendritique. On ne connaît pas non plus la relation entre les signaux d'entrée et de sortie d'une telle sous-unité, ou bien si le nombre et la taille des sous-unités peuvent être modifiés par certains motifs des signaux d'entrée. Par conséquent, on ne peut que deviner l'influence des dendrites sur le fonctionnement des réseaux de neurones.

Dans cette thèse, nous avons choisi une approche mathématique pour comprendre le fonctionnement des dendrites. Une telle approche peut aider à éclaircir le fonctionnement des dendrites car elle permet de dériver des modèles de dendrites simplifiés, mais précis, qui capturent toujours leurs effets computationnels les plus importants. Puis, ces modèles simplifiés peuvent être analysés et compris entièrement, ce qui nous apprend comment les dendrites réelles, elles, fonctionnent computationnellement. De plus, simuler ces modèles simplifiés exige typiquement moins de calculs que de simuler des modèles détaillés. Simuler des réseaux de neurones avec des dendrites en est ainsi facilité, ce qui facilite en retour l'étude de l'influence des dendrites sur des réseaux de neurones.

La fonction de Green constitue la base de notre approche théorique. D'abord, nous avons cherché à comprendre si la fonction de Green pouvait réduire le coût des simulations des dendrites. Cette fonction a une structure spéciale qui découle du fait qu'elle est définie sur un arbre. Une découverte mathématiquement intéressante dans ce contexte est que cette

Acknowledgements

structure permet de fortement réduire le nombre d'équations nécessaires pour modéliser un arbre dendritique. Néanmoins, nous avons été forcés de reconnaître que simplifier les dendrites de cette façon n'apporte pas d'information nouvelle sur l'hypothèse des sous-unités. Ensuite, nous avons redirigé nos efforts vers une autre façon de décomposer la fonction de Green. Le modèle de dendrite que nous avons construit de cette façon nous a apporté de nombreux enseignements sur l'hypothèse des sous-unités. En particulier, nous avons remarqué que l'occurrence des sous-unités était bien prédite par le ratio de l'impédance locale sur l'impédance de transfert dans les dendrites. Cette découverte a permis d'estimer le nombre de sous-unités qui peuvent coexister sur un arbre dendritique. Nous avons constaté aussi que ce ratio peut être modifié par d'autres signaux d'entrée, en particulier des conductances inhibitrices, et donc que le nombre des sous-unités sur un arbre dendritique peut être modifié d'une façon dynamique. Finalement, nous avons pu démontrer que, à cause de cette augmentation dynamique du nombre de sous-unités, des branches individuelles qui autrement auraient répondu aux signaux d'entrée comme une unité singulière, peuvent devenir sensible aux motifs de signaux d'entrée différents. Nous pensons que ce modèle peut être implémenté d'une façon très efficace. Une fois inclus dans des logiciels de simulations des réseaux de neurones, il pourrait donc faciliter considérablement l'accès aux simulations des réseaux de neurones avec dendrites pour les modélisateurs.

Mots clefs : Dendrites, Sous-unités, Modèles simplifiés, La fonction de Green

Contents

Acknowledgements	i
Abstract (English/Français)	iii
Contents	vii
List of figures	xi
1 Introduction	1
1.1 How dendrites are modeled	3
1.1.1 The cable equation	3
1.1.2 Ion channels: the Hodgkin-Huxley formalism	3
1.1.3 Synapses: linear and non-linear conductances	5
1.2 Solving dendrites	6
1.2.1 Compartmental modeling	6
1.2.2 The Green's function: from Fourier to the separation of variables	7
1.3 Simplification strategies for dendrite models	10
1.3.1 Linear response characteristics: collapsing the dendritic tree	10
1.3.2 Lumping compartments	12
1.3.3 Reduced order modeling	12
1.4 The subunit hypothesis	15
1.4.1 The significance of subunits	15
1.4.2 The need for independent dynamical variables	17
1.5 Structure of this thesis	18
2 The Green's function formalism as a bridge between single and multi-compartmental modeling	19
2.1 Introduction	19
2.2 Synapse model based on the Green's function formalism	21
2.2.1 The neuron model in time and frequency domains	22
2.2.2 Morphological simplification by applying Green's function	24
2.3 Model implementation & Validation	28
2.3.1 Synapse model implementation	28
2.3.2 Multi-compartmental and point-neuron model	28

Contents

2.3.3	Input-order detection with differential dendritic filtering	29
2.3.4	Voltage-gated active currents	31
2.3.5	Multiple synapse interactions	32
2.3.6	Runtime	32
2.4	Discussion	32
3	A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models	35
3.1	Introduction	35
3.1.1	The system of equations	37
3.2	Methods	39
3.2.1	A sparse reformulation of the Green's function formalism	39
3.2.2	An efficient method to integrate the system of Volterra integral equations	45
3.2.3	Nonlinear terms and the small Δx limit of the SGF formalism	49
3.2.4	Implementation	53
3.3	Validation	54
3.3.1	Computational cost	56
3.4	Summarizing remarks	58
3.5	Discussion	59
3.6	Appendix	61
3.6.1	Proof of Lemma 1	61
3.6.2	Vector fitting	64
4	Dynamic compartmentalization in neurons enables branch-specific learning	67
4.1	Introduction	67
4.2	Results	68
4.2.1	The neural evaluation tree	68
4.2.2	The impedance-based independence index	68
4.2.3	Dendritic compartmentalization	71
4.2.4	Dynamic compartmentalization	71
4.2.5	Branch-specific learning	75
4.2.6	Conclusion	77
4.3	Biophysical modeling	80
4.3.1	Morphologies	80
4.3.2	Physiological parameters	80
4.3.3	Plasticity	81
4.3.4	Compartmental models	82
4.3.5	Green's function and the separation of variables	82
4.3.6	Synaptic activation	83
4.4	Neural evaluation tree	84
4.4.1	Introduction: two input regions	84
4.4.2	Full neurons: deriving the impedance tree	86
4.4.3	An efficient solution algorithm	89

4.4.4	Conductance-based synapses: analytic solutions	95
4.5	Predicting spikes: linear layers	100
4.5.1	Matrix inversion with linear layers.	101
4.6	Independence and compartmentalization	103
4.6.1	True independence	103
4.6.2	Estimating independence	103
4.6.3	Compartmentalization	105
4.7	Simulation-specific parameters	106
5	Conclusion	109
	Bibliography	113
	Curriculum Vitae	129

List of Figures

1.1	A historical perspective on dendrites	4
1.2	Functions and time-scales of the SOV expansion	9
1.3	Reproducing voltage statistics with the SOV expansion.	11
1.4	Comparison between SOV and POD basis.	14
1.5	Basis of neural computation and the relevance of dendritic subunits	16
2.1	Comparison between a reference multi-compartmental model and a point-neuron model equipped with the new synapse model implicitly simulating dendritic processing.	30
2.2	Comparison between “implicit” and “explicit” model neurons of a pyramidal cell.	31
3.1	Schematic of the rationale behind the sparsification of the Green’s function formalism for tree structures.	41
3.2	Validation of the SGF formalism on axon and dendrite models.	55
3.3	The execution time of the SGF formalism compared to a NEURON simulation.	57
3.4	Illustration of the VF algorithm.	65
4.1	Constructing a neural evaluation tree allows for the systematic evaluation of crosstalk between input sites.	69
4.2	The impedance-based independence index I_Z leads to a systematic characterization of independent compartments.	72
4.3	Additional analysis for the two-branch toy model.	73
4.4	Detailed analysis of the NET approximation for three cell morphologies.	74
4.5	Dynamic compartmentalization due to spatio-temporal input patterns.	76
4.6	Tuning branch-specific learning with shunting conductances.	78
4.7	Analysis of five classes of cortical cells.	79
4.8	Cell morphologies.	80
4.9	Neural evaluation tree.	85
4.10	Illustration of the first step of the recursive algorithm to derive an NET tree.	87
4.11	Distinction between simple and complex dendritic arborizations in the first step of the NET tree generation algorithm.	89
4.12	Error of the NET approximation	90
4.13	Schematic representation of the inversion algorithm	91
4.14	Pseudocode for the up and down sweep phases of the matrix inversion algorithm.	96

List of Figures

4.15 Solution for a single synapse type.	98
4.16 <i>SL</i> in a subtree of a stellate cell	99
4.17 LinL's to improve dendrite to soma transfer.	101

1 Introduction

Since the pioneering work of Ramon y Cajal [Lerma and De Carlos, 2014], it has been known that neurons consist of a soma and axonal and dendritic arborizations. As Cajal correctly hypothesized, information flows in a directed fashion within these structures (Fig 1.1A): inputs are received on the dendrites, flow through to the soma, where an output in the form of rapid depolarizations, called action potentials (AP), is generated, that is then transmitted downstream through the axon. Although dendrites were originally believed to simply collect inputs and transmit them to the soma, evidence has emerged in the last three decades that their function is much more intricate. It was shown that they are equipped with mechanisms that allow non-linear processing of inputs [Johnston et al., 1996, Häusser et al., 2000, Häusser and Mel, 2003, London and Häusser, 2005, Spruston, 2008, Branco and Häusser, 2010, Antic et al., 2010], so that one nerve cell can effectively perform computations that were previously believed to be only possible with networks of neurons. The advent of in-vivo recording techniques in recent years has allowed for corroborating evidence, showing that non-linear dendritic processing not only happens during, but is required for normal behaviour [Grienberger et al., 2015, Takahashi et al., 2016, Smith et al., 2013, Moore et al., 2017].

The list of computations possible with dendrites is long. Indeed, dendrites enable branch-specific control of AP backpropagation [Müllner et al., 2015], a process in which the generated AP propagates back through the dendritic tree, thought to provide important feedback information to synapses to determine whether they will strengthen or not [Gerstner and Kistler, 2002]. Dendrites thus allow precise control over which synapses will learn and which synapses will remain static [Golding et al., 2002, Sjöström et al., 2008]. Dendrites also allow for the generation of local, non-linear cooperative events, called dendritic spikes, when multiple inputs arrive together [Larkum et al., 2007, Wei et al., 2001]. These events in turn are thought to allow individual neurons to function as multilayer neural networks [Poirazi et al., 2003b, Poirazi et al., 2003a, Behabadi and Mel, 2014]. Dendrites furthermore enable the direction-specific detection of input sequences [Branco and Häusser, 2010], and differentiation between different input streams based on their location on the neuron [Johanning et al., 2009] and/or their frequency components [Laudanski et al., 2014]. Moreover, dendrites are hypothesized to

increase the memory capacity of neural networks [Poirazi and Mel, 2001, Wu and Mel, 2009, Kastellakis et al., 2016].

Understanding the influence of these computations on network dynamics is of vital importance for understanding brain function. Nevertheless, barring a few exceptions [Markram et al., 2015, Egger et al., 2015], the dendritic tree is usually omitted in network modeling. We identified two reasons for this omission. First, simulating the dendritic tree is computationally expensive. Whereas simulating a neuron as a point requires integrating a few tens of first-order coupled differential equations at most, simulating dendrites in their full detail requires a few thousands of equations. Second, if one wants to include a particular dendritic computation in a neural network model, it is poorly understood how one can derive a simpler model that captures the essence of the computation. Such conceptually simplified models could not only improve the efficiency of simulating the computation in question, but also improve our understanding of the computation itself and its influence on the dynamics of networks of neurons.

During the research that led to this thesis, we aimed to advance both points. We first studied whether the Green’s function (GF) formalism [Bayin, 2006], a technique pioneered by the British mathematician George Green to solve differential equations, could be used to improve the computational models of neurons. While we made great progress in improving the GF formalism (see chapters 2 and 3, or [Wybo et al., 2013, Wybo et al., 2015]), the range of applications where this formalism provides substantial advantage over the now highly standardized compartmental models [Hines, 1984, Carnevale and Hines, 2006] remains too small for wide-spread adoption. We then turned our attention to the second point, in particular to a computational aspect of dendrites that has received a lot of attention in recent years: *the subunit hypothesis*. According to this hypothesis, inputs arriving at certain loci on the dendritic tree undergo local, non-linear processing independently from inputs at other loci. As a consequence, the rate-based response function of neurons bears similarities to a multilayer neural network [Poirazi et al., 2003b, Poirazi et al., 2003a, Behabadi and Mel, 2014]. In § 1.4, we review some evidence that this finding can substantially enhance the computational power of neural circuits. Nevertheless, the precise separation between dendritic loci required for them to function as independent subunits remains unknown. By consequence, the number of subunits that can coexist on a dendritic tree also remains elusive, as well as the input-output relation of the subunits and whether their size and number can be modified by input patterns. We hypothesized that this lack of understanding stems from a lack of accurate, but conceptually simple models that can describe these subunits as independent dynamical integrators. In chapter 4, a method to derive such conceptually simple models of neural subunits is described.

To understand (the need for) these methods to simplify and understand dendrities, it is useful to be familiar with the way dendrites are modeled (§ 1.1) and with how solutions for dendritic voltage transients can be obtained (§ 1.2). We also describe a number of existing simplification strategies for dendrites and why they fall short of elucidating the subunit hypothesis (§ 1.3).

1.1 How dendrites are modeled

Models have proved instrumental in understanding dendritic function. Many facts about dendrites that are now standard knowledge have first been predicted through the simulation and analysis of mathematical dendrite models. It is therefore useful to discuss how dendrites are modeled and how solutions for voltage transients within the dendritic tree can be obtained.

1.1.1 The cable equation

The cable equation models leaky conduction lines, conceptualized as sequences of parallel RC-circuits (Fig 1.1B, see [Dayan and Abbott, 2005] for an instructive derivation). In the 1950's, this equation had been used to successfully model the spread of currents through axonal structures. Nevertheless, little was known about its use for dendrites, until Wilfried Rall showed that by connecting the dendrites to the soma, the peculiar combination of time-scales observed in the decay of somatic voltage transients could be explained [Rall et al., 1995]. His explanation was as simple as it was elegant: instead of only leaking through the membrane, the injected current also flows axially, back into the dendritic tree (Fig 1.1C), thus inducing faster time-scales than expected in the somatic voltage transient. Rall hence showed for the first time that the dendritic tree is *not* an iso-potential unit. Elaborating on this finding, he predicted that at the thin tips of dendrites, relatively small synaptic input currents cause large depolarizations, that in turn attenuate heavily on their way to the soma. This finding, not confirmed in cortex until much later [Nevian et al., 2007], would lay the basis for the subunit hypothesis [Häusser and Mel, 2003].

In nerve fibers, the cable equations takes on the following form:

$$2\pi a c_m \frac{\partial V}{\partial t}(x, t) + \frac{\pi a^2}{r_a} \frac{\partial^2 V}{\partial x^2}(x, t) - 2\pi a g_m (V(x, t) - E_{eq}) = I(x, t), \quad (1.1)$$

where a is the fiber's radius, c_m the cell's membrane capacitance per unit area, g_m the membrane conductance per unit area and r_a the axial resistance over a unit of length. The membrane potential is denoted by $V(x, t)$ and the input current along the fiber by $I(x, t)$. At rest ($I(x, t) = 0$), the inside of the cell is found to have a negative polarity compared to the extracellular medium, captured in this framework in the equilibrium potential E_{eq} (which usually takes on values between -75 and -65 mV). To this day, equation (1.1) is used to model signal conduction in nerve fibers. Contemporary modeling efforts enrich it with ion channels and synaptic input conductances.

1.1.2 Ion channels: the Hodgkin-Huxley formalism

In 1952, Hodgkin and Huxley published a seminal paper [Hodgkin and Huxley, 1952] explaining and modeling APs in the giant axon of the squid through the interplay of Na⁺ and K⁺ permeable ion channels. They conceptualized these channels as conductances, consisting of

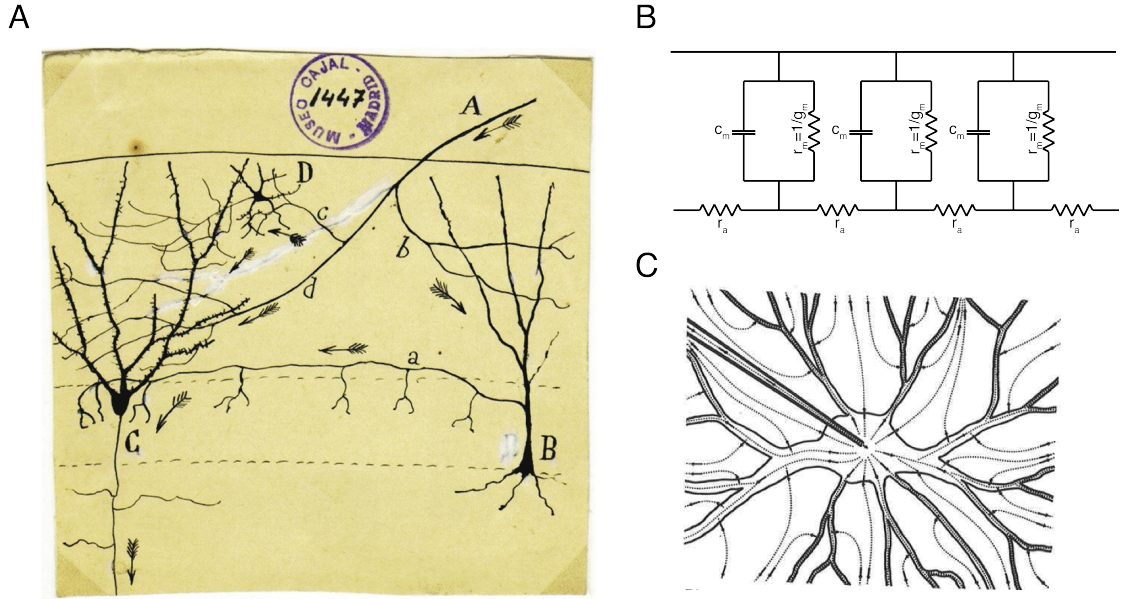


Figure 1.1 – **A historical perspective on dendrites.** **A:** Cajal’s drawing of a Golgi-stained cell network. Arrows indicate his hypothesis on the flow of information in the cell. Adapted from [Lerma and De Carlos, 2014]. **B:** Conceptual representation of the electrical circuit associated with a nerve fiber. **C:** Diagram of the current flow through a dendritic arborization upon creation of an electrical field between two electrodes, one of them in the soma (the other electrode is not shown). Current flows longitudinally through the neuronal fibers and also leaks through the cell membrane. Adapted from [Rall et al., 1995].

different units whose opening and closing dynamics depend non-linearly on the membrane potential (see [Dayan and Abbott, 2005] for an instructive description). Since then, many more ion channels have been measured and modeled [Podlaski et al., 2017], and it has been shown that they do not just occur in the soma and axon; the dendritic tree is also littered with them [Migliore and Shepherd, 2002, Angelo et al., 2007, Almog and Korngreen, 2014]. In-vivo studies furthermore suggest that these ion-channels are active during and required for normal behaviour [Takahashi et al., 2016, Moore et al., 2017].

Most often, ion channels are modeled as a set of gating variables $\mathbf{y} = (y_1, \dots, y_n)$, that each depend on the local voltage through an activation function $y_i^\infty(V)$. This function also determines the steady state value of the gating variable at a given voltage (hence it is denoted by ∞ superscript), which the variable approaches with a time-scale that may also depend on the voltage:

$$\dot{y}_i = \frac{y_i^\infty(V) - y_i}{\tau_{y_i}(V)} \quad \text{for } i = 1, \dots, n.$$

The current associated with the ion channel is then given by Ohm’s law, as the product of

conductance and potential difference:

$$I_c = \bar{g}_c f_c(\mathbf{y}) (V - E_c).$$

Here, I_c denotes the channel current \bar{g}_c the maximal conductance of the channel type (determined as the product of conductance of a single channel and the number of channels per unit area) and f_c the fraction of open channels, which depends non-linearly on the gating variables (who in turn depend non-linearly on the voltage). The factor $(V - E_c)$ serves as the driving potential of the current, where the value of E_c , the channel's reversal potential, is derived from the concentrations of the ions the channel is permeable to by thermodynamic arguments [Dayan and Abbott, 2005]. Channel currents of this form are easily integrated in the cable equation (1.1) by adding them as a term to the input current $I(x, t)$ on the right-hand side.

1.1.3 Synapses: linear and non-linear conductances

Synapses transmit information between neurons. At the presynaptic side, output spikes trigger neurotransmitter release: glutamate for excitatory synapses or gamma-Aminobutyric acid (GABA) at inhibitory synapses. On the postsynaptic neuron, glutamate binds either to alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) or N-methyl-D-aspartate (NMDA) receptors, whereas GABA binds to GABA receptors.

Both AMPA and GABA receptors are modeled by a current of the form:

$$I_s = g_s(t) (V - E_s), \quad (1.2)$$

where $g_s(t)$ is the synaptic conductance window, that models the time-course of transmitter binding (most often an alpha-function or double exponential [Rotter and Diesmann, 1999]). AMPA and GABA receptors are permeable to different ions and hence have different reversal potentials E_s : the AMPA current changes sign at around 0 mV and the GABA current around -80 mV. Hence, AMPA currents tend to depolarize the cell – they are excitatory – whereas GABA currents tend to hyperpolarize the cell – they are inhibitory. NMDA receptors, which often occur in conjunction with AMPA receptors and also have a reversal potential of around 0 mV, depend non-linearly on the voltage through a magnesium block [Jahr and Stevens, 1990], usually modeled as a sigmoidal dependence of the conductance on voltage:

$$I_s = g_s(t) \sigma(V) (V - E_s), \quad (1.3)$$

where the following function is often used for the sigmoid:

$$\sigma(V) = \frac{1}{1 + e^{-\frac{V-12}{10}}}. \quad (1.4)$$

From this functional form, it can be seen that the sigmoid will be close to zero around the equilibrium potential $E_{eq} \simeq -70$.

In both cases, the synaptic current depends on the local voltage. Nearby synapses may exert influence on this voltage, and hence collectively modify the synaptic current. In particular, nearby synapses may aid in depolarizing a stretch of dendrite, thus removing the magnesium block of the NMDA receptors, resulting in a much larger current and dendritic depolarization (the sigmoid (1.4) will have a much larger value). This phenomenon is known as the NMDA spike [Rhodes, 2006, Major et al., 2008, Antic et al., 2010]. Together with the strong attenuation in dendrites, this phenomenon forms the basis for the subunit hypothesis. Synaptic currents can be added straightforwardly to the right-hand side of the cable equation (1.1). They are localized in space, and are thus multiplied with a spatial Dirac-delta function $\delta(x - x_s)$, where x_s is the spatial coordinate of the synapse.

1.2 Solving dendrites

1.2.1 Compartmental modeling

After proposing the cable equation to model dendrites, Rall obtained solutions for transient synaptic inputs by sub-dividing the fiber in a discrete number of compartments [Rall et al., 1995]. Mathematically, this amounts to replacing the double spatial derivative in (1.1) by its 2nd order finite difference (FD) approximation:

$$\frac{\partial^2 V}{\partial x^2}(x_i, t) \approx \frac{V(x_{i-1}, t) - 2V(x_i, t) + V(x_{i+1}, t))}{\Delta x^2}, \quad (1.5)$$

where x_{i-1} , x_i and x_{i+1} are points on the fiber mutually separated by a distance Δx . Upon substituting this in (1.1), and rearranging terms, one finds:

$$\begin{aligned} & 2\pi a \Delta x c_m \frac{\partial V}{\partial t}(x_i, t) + \\ & \frac{\pi a^2}{r_a \Delta x} (V(x_{i-1}, t) - V(x_i, t)) + \frac{\pi a^2}{r_a \Delta x} (V(x_{i+1}, t) - V(x_i, t)) - \\ & 2\pi a \Delta x g_m (V(x_i, t) - E_{eq}) = \Delta x I(x_i, t). \end{aligned} \quad (1.6)$$

It can be seen that this equation is a literal implementation of the circuit depicted in Fig 1.1B: each compartment is equipotential with voltage $V(x_i, t)$, has a capacitance of $2\pi a \Delta x c_m$ and a resistance of $1/2\pi a \Delta x g_m$ and is coupled to the neighboring compartments by an axial resistance of $r_a \Delta x / \pi a^2$. Hence, a compartmental model is a large system of coupled ordinary differential equations (ODE).

Solving such a system in a stable manner, especially when ion channels and non-linear synapses are involved, requires inverting a matrix of the same dimension as the number L of locations at which the voltage is evaluated [Joyner et al., 1978]. In non-branching nerve fibers, where this matrix is tri-diagonal, this operation is feasible as it can be carried out in $O(L)$ steps [Moore et al., 1978]. Simulating more complex arborizations was prohibitively expensive in the early days of compartmental modeling, since normal matrix inversion requires $O(L^3)$ steps.

Hines however showed that because neuronal arborizations can be conceptualized as *tree graphs*, an inversion algorithm can be derived that still only requires $O(L)$ steps [Hines, 1984]. He incorporated this algorithm in his NEURON simulator [Carnevale and Hines, 2006], which to this day is the most used simulation program to model dendrites.

Despite the great success of compartmental modeling, many questions still remain, as understanding simulation results is often a tedious and time-consuming process. A complementary, theory-driven approach thus remains a necessity [Gerstner et al., 2012], as conceptually simple models that still capture the phenomena of interest, aid greatly in elucidating them. That is why, during this thesis, the lack of concrete answers on many questions regarding the subunit hypothesis nudged us towards investigating the GF formalism.

1.2.2 The Green's function: from Fourier to the separation of variables

Before Hines discovered his algorithm that made compartmental simulations so successful, much research was dedicated to deriving analytical steady-state and transient solutions [Jack et al., 1975, Tuckwell, 1988] on dendritic trees. Such approaches often employ the GF $G(x, x', t - t')$, which is obtained as the solution of (1.1) when the right hand side is a Dirac delta pulse ($I(x, t) = \delta(x - x')\delta(t - t')$). When the GF is found, the solution for an arbitrary input is obtained by convolving that input with the GF:

$$V(x, t) = \int dx' dt' G(x, x', t - t') I(x', t'). \quad (1.7)$$

The GF thus contains all the information on how input current is converted into output voltage. Butz and Cowan were the first to obtain the GF for any dendritic structure by deriving a set of graphical rules that allowed for its computation in the Fourier domain [Butz and Cowan, 1974]. However, because of its intricate functional form and the difficulty of obtaining expressions in the time domain, this approach remained of limited usefulness until Koch used it to derive a computer algorithm to numerically evaluate the GF [Koch and Poggio, 1985]. Koch then used the GF to analyze several computational aspects of synaptic interactions in dendrites [Koch et al., 1982, Koch et al., 1983]. Note that his algorithm is not limited to equations of the form (1.1); it can also incorporate linearized ion channels [Mauro et al., 1970, Koch, 1984].

Nevertheless, Koch later abandoned the GF, stating the drawbacks “(1) *In the presence of N current inputs – rather than a single one – on the order of N^2 additional computations have to be performed; that is, these methods do not at all scale well to massive synaptic input.* (2) *While these methods can be adapted to treat synaptic input as conductance changes, it is not computationally efficient to do so.* (3) *Finally, and fatally, they assume linearity and fail in the presence of voltage-dependent membrane components*” [Koch, 1998]. We were however able to show, using linear algebra and the tree graph structure of the neuronal arborization, that these three points could be overcome by rearranging the system of GF equations [Wybo et al., 2015]. The obtained formalism can outperform standard compartmental simulations in certain use-cases. Nevertheless, while our approach is mathematically interesting, the lack of readily

available simulation software as well as the relatively small range of use cases impedes its wide-spread adoption.

Both with Kochs method and our rearranged version, understanding where and under what conditions subunits arise remains intractable. We therefor turned our attention to the separation of variables (SOV) approach, where the GF is approximated by a series of the form:

$$G(x, x', t - t') = \sum_{k=0}^{\infty} \phi_k(x) \phi_k(x') e^{-\frac{t-t'}{\tau_k}}. \quad (1.8)$$

In this expression, the time-scales τ_k decrease for increasing k and the functions ϕ_k are combinations of sine functions whose spatial scales decrease for increasing k . While Rall and others had already derived solutions of this form for simple morphologies [Holmes et al., 1992], it was Major who, in a mathematical tour de force, found an algorithm that could compute such solutions for arbitrary dendritic trees [Major et al., 1993a, Major et al., 1993b, Major, 1993, Major and Evans, 1994].

Being an infinite series, it is important to find a good estimate for the maximal number K of terms that have to be included to accurately reproduce the membrane potential. Following equation (1.7), we found that the overall importance Υ of a term in this series for the voltage dynamics is well indicated by:

$$\Upsilon = \tau_k \left[\int dx |\phi_k(x)| \right]^2.$$

As this value is proportional to τ_k , it was our hope that if we could find a minimal time-scale τ_K beyond which the series could be truncated, the associated minimal spatial scale would provide us with the size of the dendritic subunits.

This undertaking was bound to fail for two reasons. First, the functions ϕ_k are global; they integrate inputs from across the whole neuron. As can be seen in Fig 1.2, independent dynamics are implemented by compensatory functions ϕ_k . In a two branch case for instance (Fig 1.2A), one function will have the same sign in both branches, whereas another function (with similar time- and thus spatial scale) will have different signs. The overall consequence being that if an input arrives in one branch, both functions will sum to construct the voltage there, whereas in the other branch they will annihilate. When more branches are added, the number of compensatory functions needed to be able to annihilate the voltage in all other branches is equal to the total number of branches (Fig 1.2B,C). In real morphologies, with complex branching patterns, the situation is not clear-cut and searching for all compensatory functions becomes difficult (Fig 1.2D). Second, while sets of compensatory functions could in theory provide some idea of the broad location of the subunits, many more smaller time-scale terms have to be included in series (1.8) to accurately reproduce the local dynamics. There may be a large grey zone where terms both contribute to subunit construction and to the accurate reproduction of the local dynamics. Fig 1.3 illustrates this point: the main

1.2. Solving dendrites

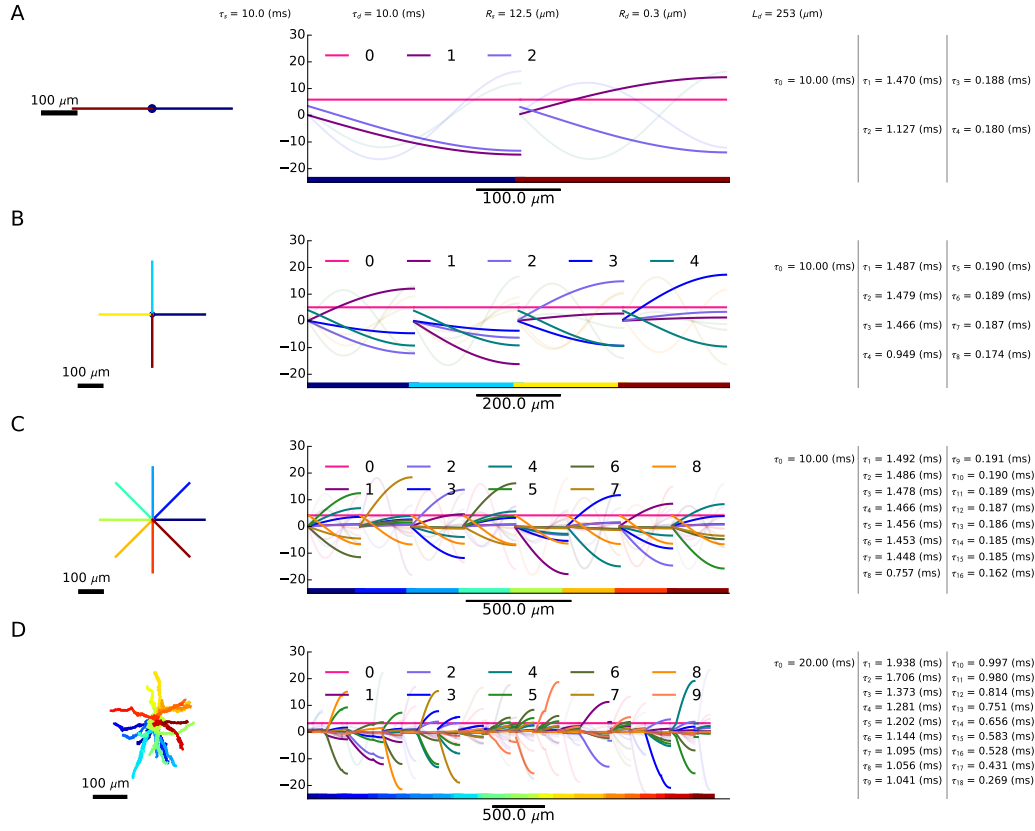


Figure 1.2 – Functions and time-scales of the SOV expansion. Biophysical dendrite parameters are shown on top. **A, B, C:** SOV-expansion for resp. 2, 4 and 8 branch models. The left panel shows the morphology, different branches have different colors. Middle panel shows the functions ϕ_k for $k = 0$ to B (with B the number of branches). The functions for $k = B + 1$ to $2B$ are also shown (transparent). Color on the x-axis corresponds to the color of the associated branch in the left panel. The right panel shows the associated time-scales τ_k which can be sub-divided in 3 groups according to their value: $k = 0$ is always the membrane time-scale, $k = 1$ to B is the first group of compensatory functions and $k = B + 1$ to $2B$ fixes the spatial voltage distribution within each branch. **D:** SOV expansion for a stellate cell morphologies. Left, middle and right panels are simular as in A, B and C, but groups of time-scales can not be distinguished.

compensatory function have indices $k = 1, 2$ (Fig 1.3B). Upon stimulating the tips of both branches with independent Ornstein-Uhlenbeck processes, it can be seen that these terms are responsible for the decrease in voltage correlation between the blue branch and location x_{i_1} (Fig 1.3F). Nevertheless, the terms with indices $k = 3, 4$ are still required to approximate the voltage statistics reasonable well (Fig 1.3D-F).

By consequence, we refrained from using the SOV-approach to understand dendritic subunits. This approach proved very useful nonetheless, as it expresses the GF in a convenient form: all the information required to compute the GF can be stored in a vector of time-scales τ_k of size K and a matrix $\Phi_{kl} = \phi_k(x_l)$ of size $K \times L$ (with L the number of locations at which the GF needs to be evaluated). Furthermore, if one wants to simulate with the GF, the convolution (1.7) becomes a sum of convolutions with exponentials, which in turn can be solved as simple ODEs [Wybo et al., 2015]. We thus used the SOV method often throughout this thesis. It should be noted though, that while not fundamentally impossible, incorporating linearized ion channels would be very challenging. To compute the time-scales τ_k , one constructs a transcendental function and solves for its zeros. In passive morphologies, these zeros are all real numbers. With linearized ion channels, these zeros may be complex conjugate pairs that lie anywhere in the negative half of the complex plane. While algorithms exist to compute these zeros [Davies, 1986, Kravanja and Van Barel, 2000c, Kravanja and Van Barel, 2000a, Kravanja and Van Barel, 2000b], the extremely complex form of the transcendental function for real morphologies, with infinite numbers of closely spaced poles and zeros, makes it very challenging to find a subdivision of the complex plane that allows the zero-finding algorithms to be well-conditioned and work successfully.

1.3 Simplification strategies for dendrite models

Efforts to simplify dendrite models, whether it is to gain a better understanding of how dendrites function or whether it is to gain computational efficiency in computer simulations, are as old as these models themselves. Here, we explore three approaches that have been commonly used in the past decades.

1.3.1 Linear response characteristics: collapsing the dendritic tree

When a synaptic current is injected somewhere along the dendritic tree, and the response is measured at the soma, two things happen: the somatic response is slightly delayed with a broader waveform compared to the dendritic current pulse and its amplitude is lower than the dendritic depolarization [Nevian et al., 2007]. These effects were predicted by Rall [Rall et al., 1995] and are respectively referred to as ‘delay’ and ‘attenuation’. Rall also realized that for artificial dendritic trees with certain symmetry constraints, the exact same response characteristics could be obtained by appropriately mapping the dendritic tree onto a single cylinder [Rall et al., 1995]. This operation can be construed as the first simplification strategy for dendritic trees. While the restrictive constraints of Rall were later relaxed somewhat

1.3. Simplification strategies for dendrite models

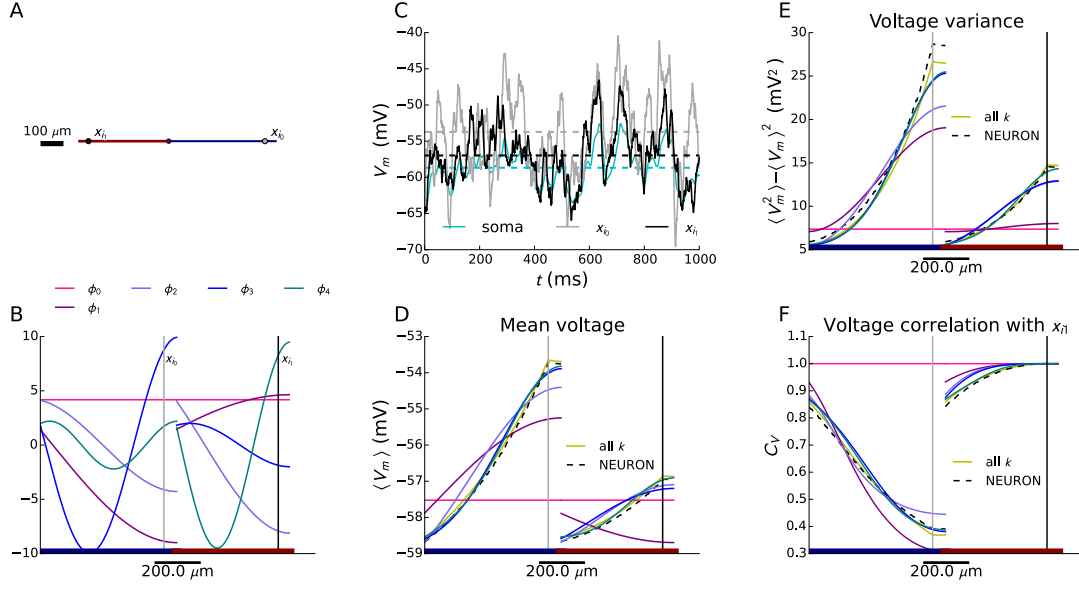


Figure 1.3 – Reproducing voltage statistics with the SOV expansion. **A:** A simple bipolar morphology. Locations x_{i_0} and x_{i_1} , where an Ornstein-Uhlenbeck current is injected, are indicated. **B:** The SOV functions ϕ_k ($k = 0, \dots, 4$) associated with the morphology. Locations x_{i_0} and x_{i_1} are indicated with a vertical line. **C:** The membrane voltage at both locations and at the soma (average value is indicated with a dashed line). **D:** The mean of the voltage at all locations, once computed with NEURON (black, dashed) and once with a large number of SOV terms (yellow). Further lines indicate the mean computed with few SOV terms, where the color corresponds to the same color in B and indicates that all terms were used for which $k \leq$ the index of the corresponding ϕ . **E:** Same as D, but for the variance. **F:** Same as D, but for the correlation with location x_{i_1} .

[Ohme and Schierwagen, 1998], real morphologies still do not adhere to these constraints and consequently this approach remains of limited use.

Modern methods take a more ‘fuzzy’ approach to reproducing the linear response characteristics of dendrites in simplified models. They content themselves with only reproducing these characteristics approximately, but with the benefit that the methods works for any dendritic tree. Essentially, the quality of any linear reduction can be measured by how well it approximates the GF from dendrite to soma $G(x = 0, x', t - t')$. A simple but reasonably accurate approach would be to define a number of different domains on the dendritic tree (which will depend mainly on the electrical distance to the soma), define an average kernel for each domain and then filter all inputs arriving in that domain with this kernel. Further computational efficiency could be gained by approximating this kernel with a sum of exponentials (for instance by using the vector fitting algorithm [Gustavsen and Semlyen, 1998, Gustavsen and Semlyen, 1999, Hendrickx and Dhaene, 2006, Wybo et al., 2015]), so that the filtering operations can be carried out by integrating simple ODEs [Wybo et al., 2015]. Even when each kernel is comprised of only a single exponential, surprisingly accurate results can

be achieved [Rössert et al., 2016].

It should be noted though that any such method forgoes the ability to model local interactions between dendritic, voltage-dependent input currents (such as ion channels or synaptic inputs). The response at the soma in a passive neuron following injection of two current pulses will always be the linear sum of both individual responses (in a passive morphology), no matter whether these pulses are delivered to different dendritic branches or to the same branch. The current following synaptic conductance inputs however depends on the local voltage, as can be seen in equations (1.2) and (1.3), and thus the eventual somatic response will depend greatly on whether these synapses are located on the same branch or not [Hao et al., 2009].

1.3.2 Lumping compartments

Another set of methods to simplify dendritic trees is concerned with reducing the number of points in the 2nd order FD approximation at which the voltage is evaluated, or, in other words, reducing the number of compartments in (1.6). In such work, often the number of dendrites is strongly reduced too, so that only on the order of ~ 10 to 100 compartments remain [Traub and Miles, 1991, Bush and Sejnowski, 1993, Migliore et al., 1999, Traub et al., 2005], sometimes even as few as two compartments are retained [Clopath et al., 2007, Urbanczik and Senn, 2014, Chua et al., 2015].

While this approach is very convenient, as standard simulation software such as NEURON can be used, the methodology in choosing the reduced cells' parameters varies greatly among different studies and is often quite arbitrary. Sometimes the reduced model is optimized to reproduce either some experimentally recorded data or some simulation results of complex models, but often the reduction relies entirely on the needs and best judgment of the modeler. Recently, a more systematic approach was pioneered to reduce the morphology, while rescaling morphological and electrical parameters to keep the response properties stable [Marasco et al., 2012, Marasco et al., 2013]. Nevertheless, reducing the morphology in this way suffers from the same problem as described in § 1.3.1: local interactions between synapses are not captured accurately. Thus, while this strategy reduces the computational complexity of dendrite models, it does not elucidate the subunit hypothesis.

1.3.3 Reduced order modeling

Another sets of methods, referred to as 'reduced basis' (RB) methods, stems from recent efforts in the applied mathematics and engineering fields to reduce the cost of solving partial differential equations (PDE) [Quarteroni et al., 2016]. As discussed in § 1.2.1, solving for the dendritic voltage requires inverting a matrix of size L :

$$\mathbb{A}(\mathbf{g}_{\text{syn}}) \mathbf{V} = \mathbf{I}(\mathbf{g}_{\text{syn}}), \quad (1.9)$$

where $\mathbb{A}(\mathbf{g}_{\text{syn}})$ and $\mathbf{I}(\mathbf{g}_{\text{syn}})$ are the FD matrix and vector [Koch, 1998, appendix by Zador and Pearlmuter], which both depend on the synaptic input conductances. RB methods aim to express the voltage vector in a suitable basis of size $N \ll L$:

$$\mathbf{V} = \sum_{n=1}^N v_n \mathbf{U}_n = \mathbb{U} \mathbf{v},$$

where $\mathbf{v} = (v_1, \dots, v_N)$ and $\mathbb{U} = (\mathbf{U}_1 | \dots | \mathbf{U}_N)$, so that the solution can be obtained by solving the following smaller system of size N :

$$\mathbb{A}'(\mathbf{g}_{\text{syn}}) \mathbf{v} = \mathbf{I}'(\mathbf{g}_{\text{syn}}),$$

where $\mathbb{A}'(\mathbf{g}_{\text{syn}}) = \mathbb{U}^T \mathbb{A}(\mathbf{g}_{\text{syn}}) \mathbb{U}$ and $\mathbf{I}'(\mathbf{g}_{\text{syn}}) = \mathbb{U}^T \mathbf{I}(\mathbf{g}_{\text{syn}})$. These methods aim to find an optimal basis that minimizes the error between the reduced solution $\mathbb{U} \mathbf{v}$ and the full solution \mathbf{V} . Earlier approaches used empirical orthogonal functions as a basis [Monahan et al., 2009] or principal interaction and oscillation patterns [Hasselmann, 1988]. These methods generally construct a matrix of a set of well-chosen ‘snapshots’ of the dynamics (through solving (1.9) for a well chosen sampling of the input space \mathbf{g}_{syn}) and then derive a set of basis functions from this matrix according to some optimality criterion. It can be shown that using the singular value decomposition of the snapshot matrix to construct the basis, yields a minimal mean squared error between each snapshot and the subspace spanned by the basis functions [Quarteroni et al., 2016]. This popular approach, commonly referred to as the ‘proper orthogonal decomposition’ (POD), was applied by [Kellems et al., 2010] to reduce neuron models. Earlier, the same authors had also employed another approach to construct the reduced basis [Kellems et al., 2009].

Such methods come with substantial difficulties however. As discussed in § 1.2.1, solving system (1.9) requires only $O(L)$ steps. On the other hand, the reduced matrix \mathbb{A}' is dense, and thus requires $O(N^3)$ steps to invert. This may negate much of the advantage of the lower dimensionality of the system. In engineering applications this trade-off usually pays off, as the PDEs of interest are often defined on 2- or 3-dimensional domains. By consequence, many more grid-points are required to evaluate the field of interest ($L \sim 10^4$ to $L \sim 10^6$ instead of $L \sim 10^2$ to $L \sim 10^3$) and the matrix inversion in the full system requires much more than $O(L)$ steps. Furthermore, when the dendritic tree is littered with closely spaced synaptic conductances, or ion channel conductances that depend non-linearly on the voltage, the matrix $\mathbb{A}(\mathbf{g}_{\text{syn}})$ still has to be computed in the original system of size L . When dealing with such parameter dependence, RB methods assume that an affine decomposition of the matrix $\mathbb{A}(\mathbf{g}_{\text{syn}})$ and vector $\mathbf{I}(\mathbf{g}_{\text{syn}})$ exists [Quarteroni et al., 2016]:

$$\begin{aligned} \mathbb{A}(\mathbf{g}_{\text{syn}}) &= \sum_{m=1}^{M_a} \theta_m^{(a)}(\mathbf{g}_{\text{syn}}) \mathbb{A}_m \\ \mathbf{I}(\mathbf{g}_{\text{syn}}) &= \sum_{m=1}^{M_i} \theta_m^{(i)}(\mathbf{g}_{\text{syn}}) \mathbf{I}_m, \end{aligned} \tag{1.10}$$

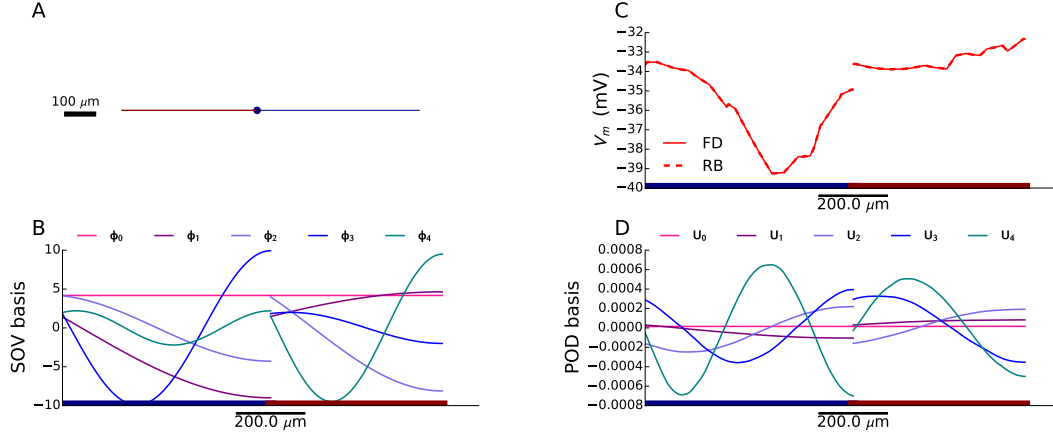


Figure 1.4 – **Comparison between SOV and POD basis.** **A:** A simple morphology with 2 cylindrical dendrites was used to compare both bases. **B:** The first five functions ϕ_k of the SOV basis for this morphology. **C:** A comparison between the steady-state FD and RB solutions for the spatial voltage distribution for a particular combination of excitatory and inhibitory input. They agree very well. **D:** The first five basis vectors \mathbf{U}_k obtained by the POD method.

with M_a and $M_i \ll L$. Consequently, $M_a + M_i$ functions θ would have to be evaluated to construct $\mathbb{A}(\mathbf{g}_{\text{syn}})$ and $\mathbf{I}(\mathbf{g}_{\text{syn}})$. In neuroscience however, any synaptic conductance is a priori independent from all others and an affine representation simply has the same number L of components as the original system. Hence, at every time-step, the full $L \times L$ matrix $\mathbb{A}(\mathbf{g}_{\text{syn}})$ and vector of size L $\mathbf{I}(\mathbf{g}_{\text{syn}})$ have to be computed. In [Kellems et al., 2010], this problem is circumvented by deriving an optimal sampling of space in order to keep L minimal. While this may yield an optimal distribution of evaluation points on the dendritic arborization, it shows that computing the full spatial voltage still remains a necessity to accurately reproduce the synaptic interactions.

We nevertheless explored whether the RB vectors could yield some information on the subunit hypothesis. To do so, we implemented a steady-state version of the FD approximation (i.e. where $\frac{\partial V}{\partial t} = 0$), and used it to generate 200 snapshots for random combinations of excitatory and inhibitory input on a simple two-branch morphology (Fig 1.4A). The POD basis obtained in this way could accurately reproduce the spatial voltage distribution (Fig 1.4C) for random excitatory and inhibitory conductances. We then plotted the first five obtained basis vectors (Fig 1.4D). As can be seen, these basis vectors bear a striking resemblance to the SOV basis (Fig 1.4B): the zero'th vector has the same value everywhere, vectors 1 and 2 have a similar shape to the SOV compensatory functions and vectors 3 and 4 have approximately the same spatial frequency as the corresponding SOV functions. Consequently, we concluded that we could not learn anything new from these RB-methods regarding the subunit hypothesis.

1.4 The subunit hypothesis

Not long after it was discovered that synaptic NMDA currents in the central nervous system depend on the local voltage [MacDonald and Wojtowicz, 1982, Mayer et al., 1984], it was hypothesized that these currents, in combination with the dendritic tree, could expand the computational repertoire of neurons [Mel, 1993], and that as a consequence the computational power of a single nerve cell would depend on its dendritic branching pattern [Schaefer et al., 2003]. Specifically, such a current could compute a non-linear transformation on the inputs arriving on a local stretch of dendrite. Hence, this stretch of dendrite would be a computational ‘subunit’ of the neuron. Together, these subunits would add extra ‘layers’ of non-linear processing to the brain [Häusser and Mel, 2003], a finding especially relevant today given the success of recent deep-learning efforts [Mnih et al., 2015, Silver et al., 2016].

1.4.1 The significance of subunits

Broadly speaking, in a neuron with D dendritic subunits (indexed by d), which each receive N_d inputs x_n , the rate-based output O can be summarized as [Poirazi et al., 2003b, Poirazi et al., 2003a]:

$$O = h_s \left(\sum_{d=0}^D h_d \left(\sum_{n=0}^{N_d} x_n \right) \right), \quad (1.11)$$

with h_s the somatic non-linear activation function and h_d the non-linear activation function of subunit d . Contrastingly, the model used in most computational studies rather has the following form:

$$O = h_s \left(\sum_{d=0}^D \sum_{n=0}^{N_d} x_n \right). \quad (1.12)$$

As can be seen, the dendritic non-linearity is omitted and the somatic output is just a function of the sum of all inputs. This difference in the functional form of the input-output relation has drastic consequences for the computational power of a nerve cell.

To understand why this is so, one must first understand a few basic principles of how neurons compute. In 1962, Hubel and Wiesel published a seminal paper on the functional organization of the cat’s visual cortex [Hubel and Wiesel, 1962]. Upstream of the visual cortex, in the lateral geniculate nucleus, they found cells that responded to a dot of light at a given location in the cats’ receptive field. In the visual cortex, they found cells that were selective to a line of light at a given location in the receptive field (Fig 1.5A). They hypothesized that if a cortical neuron would receive inputs from multiple co-linear lateral geniculate cells, it could be excited above its AP threshold and activate. Hence, this neuron would be sensitive to a line of light. When multiple such neurons, all sensitive to for instance a vertical line at some coordinate in the receptive field, would target another neuron, and this neuron would be activated above its AP

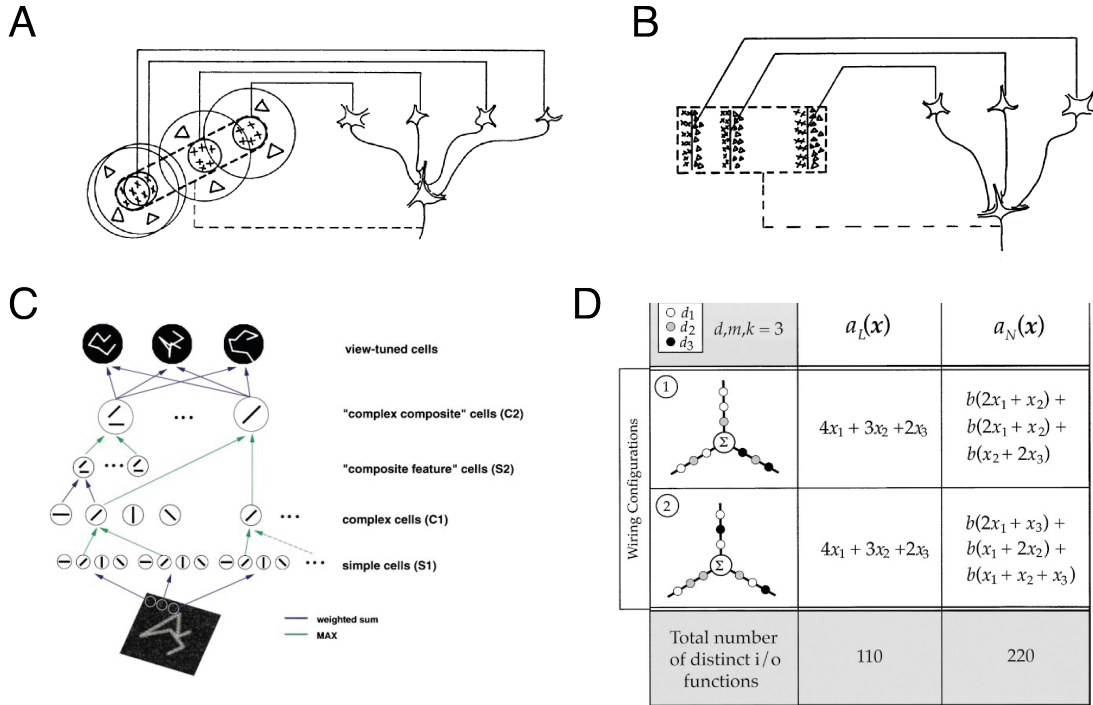


Figure 1.5 – **Basis of neural computation and the relevance of dendritic subunits.** **A:** To create a line-sensitive cell, multiple co-linear cells sensitive to a dot stimulus need to be active together to activate the downstream neuron (adapted from [Hubel and Wiesel, 1962]). **B:** To create a translation-invariant cell, the downstream neuron needs to be activated by any one of the line sensitive cells (adapted from [Hubel and Wiesel, 1962]). **C:** Today, these ideas are still pervasive in theories of object recognition (adapted from [Riesenhuber and Poggio, 1999]). **D:** subunits increase the memory capacity of neurons. A neuron with three dendrites is targeted by four white inputs (x_1), three grey inputs (x_2) and two black inputs (x_3) in two different configurations. With input-output relation (1.12), this neuron could not distinguish these configurations, whereas with input-output relation (1.11), it could (adapted from [Poirazi and Mel, 2001]).

threshold by any one of the vertical line neurons, it would be sensitive to the orientation of the line but not to its position (Fig 1.5B). These operations are nowadays still fundamental to theories of visual object recognition (Fig 1.5C) and require non-linear transformations with the right gain and threshold at each cell [Riesenhuber and Poggio, 1999]. In this way, neurons deeper in the processing stream are able to respond to more and more abstract concepts, independent of their location in the visual field. With dendritic subunits, one cell could execute a multitude of these operations at once [Mel et al., 1998, Archie and Mel, 2000].

Another question central to computational neuroscience is how neurons store information [Chaudhuri and Fiete, 2016]. Traditionally models of information storage treat neurons as equipotential units that simply sum their inputs, as in equation (1.12) [Gerstner and Kistler, 2002, Mongillo et al., 2008, Zenke et al., 2015, Brunel, 2016]. Converging experimental evidence however suggests that information is stored in a highly localized fashion on dendrites [Losonczy et al., 2008,

Makara et al., 2009, Govindarajan et al., 2011, Weber et al., 2016]. As illustrated in Fig 1.5D, much more information can in principle be stored by a neuron with subunits (1.11) than by an equipotential neuron (1.12) [Poirazi and Mel, 2001]. Only recently researchers have started to explore how this could improve information storage in networks of neurons [Kastellakis et al., 2016].

1.4.2 The need for independent dynamical variables

Despite the large number of papers that assume some form of dendritic subunits, to this day very little is understood about them. Only efforts to fit rate-based models of the form (1.11) to the biophysical neuron models have been successful [Poirazi et al., 2003b, Poirazi et al., 2003a, Behabadi and Mel, 2014]. Nevertheless, the precise timing of input and output spikes is also believed to play an important role in how the brain encodes information [Gerstner et al., 1996, Butts et al., 2007, Nemenman et al., 2008, Hong et al., 2012, Ratté et al., 2013]. A major question is thus to understand the dynamical input-output transformation that a subunit performs. Branco suggested that subunits in the dendritic tips would favor rate-based coding [Branco and Häusser, 2011], but that assumes that the excitatory input there is strong enough to remove the magnesium block of the NMDA channels. Before learning, inputs may not be strong enough to remove this block and coincident inputs might still be required to activate NMDA channels.

Another question is where on the dendritic tree these subunits are located. Clustering algorithms have failed to produce convincing results [Rabinowitch and Segev, 2006], as they rely on arbitrarily defined threshold correlations in the local membrane potential, while these correlations depend heavily on the dynamical regime the neuron is in. This problem is further complicated by the fact that subunits can not be seen independent of the inputs one aims to study. It is the location of these inputs on the morphology that determines whether they will belong to the same subunit or not.

Given this information, and after our various failed attempts to find a convincing method to determine dendritic subunits with existing mathematical method, we came to the conclusion that a method was needed that could, for any morphology and spatial combination of synapses, derive a dynamical model that (i) accurately models the spatio-temporal voltage within the dendritic tree and that (ii) features dynamical variables that only depend on the inputs arriving to the respective subunits – if and only if such subunits exist. That way, both the question about where on the dendritic tree the subunit is located and the question about how the subunit integrates inputs can be answered. In chapter 4 we describe a model that does just this, and an algorithm that derives this model for any morphology and set of synapse locations. To our surprise, this model also showed that certain types of input can actively modify the distribution of subunits on the dendritic tree. This amounts to modifying the grouping in equation (1.11) by increasing D , and thus could be another mechanism that enhances the computational repertoire of neurons.

1.5 Structure of this thesis

The structure of this thesis consist of three main parts. In chapter 2, published as [Wybo et al., 2013], we assess the computational performance of naively using the GF formalism to simulate neuron models. We show that the somatic AP can be successfully reproduced in a neuron performing an input-order detection task. We show furthermore that using the GF formalism yields computational gain for a low number of input sites on the dendritic tree. Although this number was too low for real world use-cases, chapter 2 provides a successful proof of concept that the GF formalism can be used to accurately model any dendritic tree – no matter how morphologically complex.

In chapter 3, published as [Wybo et al., 2015], we describe how the GF formalism can be sparsified, greatly reducing the number of GF kernels, and how the remaining convolutions can be integrated efficiently. With these two innovations, the number of input locations for which computational gain is achieved is augmented greatly. Thus, as described in the discussion section of chapter 3, there are real-world use-cases in which there is a gain in computational performance.

Nevertheless, with these two approaches, we found that the two objectives – a *computational simplification*, augmenting computational efficiency, and a *conceptual simplification*, rendering it easier to understand how dendrites compute – were not aligned. In particular, these approaches provided little help in understanding the subunit hypothesis. By consequence, we sought and found another approach, where the GF associated with a morphology is approximated by a tree-like, impedance based network. We found that the resulting models, which we termed neural evaluation trees (NETs), yielded much insight in the subunit hypothesis. In chapter 4 we describe the results of this approach.

2 The Green's function formalism as a bridge between single and multi-compartmental modeling

Willem A. M. Wybo, Klaus M. Stiefel and Benjamin Torben-Nielsen, published in *Biological Cybernetics* [Wybo et al., 2013].

Abstract

Neurons are spatially extended structures that receive and process inputs on their dendrites. It is generally accepted that neuronal computations arise from the active integration of synaptic inputs along a dendrite between the input location and the location of spike generation in the axon initial segment. However, many application such as simulations of brain networks, use point-neurons –neurons without a morphological component– as computational units to keep the conceptual complexity and computational costs low. Inevitably, these applications thus omit a fundamental property of neuronal computation. In this work, we present an approach to model an artificial synapse that mimics dendritic processing without the need to explicitly simulate dendritic dynamics. The model synapse employs an analytic solution for the cable equation to compute the neuron's membrane potential following dendritic inputs. Green's function formalism is used to derive the closed version of the cable equation. We show that by using this synapse model, point-neurons can achieve results that were previously limited to the realms of multi-compartmental models. Moreover, a computational advantage is achieved when only a small number of simulated synapses impinge on a morphologically elaborate neuron. Opportunities and limitations are discussed.

2.1 Introduction

Neurons are morphological structures: they have dendritic branches on which most inputs are received and an axonal tree through which the output signal is communicated with other neurons. In this light, neuronal computations can be seen as the integration of synaptic inputs along the dendrites up to the axon initial segment where an output signal is generated.

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

Hence a key role in neuronal computation is taken by the exact shape and composition of dendrites. Indeed, it is known that the neuronal response is shaped by the precise location and activation pattern of synapses [Branco and Häusser, 2010, Torben-Nielsen and Stiefel, 2010, Gidon and Segev, 2012] and by the expression and distribution of (voltage-gated) ion-channels [Migliore and Shepherd, 2002, Magee, 1999, Torben-Nielsen and Stiefel, 2010, Spruston, 2008].

Despite this proven importance, dendritic processing is usually ignored in network simulation [Gewaltig and Diesmann, 2007, Brette et al., 2007, Richert et al., 2011], but see [Markram, 2006] for an exception. One reason is the computational cost associated with multi-compartmental simulations: a costs that, at the level of the model neuron, scales with the morphological complexity of the dendritic arborization. Related is the conceptual cost associated with building detailed single-neuron models [Hay et al., 2013] with the spatial distribution of conductances across the membrane and localized non-linearities. The key is to capture the somatic voltage in response to synaptic inputs on the dendrites. Is there an alternative to multi-compartmental models to simulate the effects of dendrites on synaptic potentials, without large computational overhead?

To this end, two strategies are commonly adopted in the literature. The first consist of performing a morphological reduction by reducing the number of dendritic segments while attempting to capture crucial characteristics of dendritic processing [Traub et al., 2005, Kellems et al., 2010]. A second strategy is to by-pass multiple (dendritic) compartments altogether by using point-neurons and fit voltage-kernels that matches the dendritic signal transformation shaping the voltage waveform caused by a synaptic input at the soma [Jolivet et al., 2004, Gütig and Sompolinsky, 2006]. The fitted [Jolivet et al., 2004] or learned [Gütig and Sompolinsky, 2006] kernel is then simply added to the somatic membrane potential. While this strategy is computationally efficient and some temporal effects of dendritic processing can be captured, it is a rather crude approximation of what dendritic integration stands for and elementary features of dendritic processing, such as local interaction between inputs, are impossible to achieve.

In this work we present a true alternative based on applying the Green's function formalism to cable theory. This way we can exactly compute the effect of synaptic inputs located in the dendrites on the somatic membrane potential [Koch and Poggio, 1985]. By design we thus compute the linear transfer function between the site of the synaptic inputs and the soma. The main advantage of this approach is that the effect of synaptic inputs along a dendrite on the somatic membrane potential can be calculated analytically. Consequently, simulations in our model are independent of the morphological complexity and a full reduction to a point-neuron can be used, as the entire effect of the morphology is captured in a transfer function. This property sets our approach apart from existing methods to model dendrites implicitly: the approach based on the equivalent cable works only with geometrically tightly constrained morphologies [Ohme and Schierwagen, 1998], while, as in [Van Pelt, 1992] all branch points of a dendritic tree have to be modeled explicitly. Because we capture arbitrary dendritic morphologies by means of transfer functions, our synapse model is able to use dendrite-specific mechanism of computation, such as delay lines (as [Gütig and Sompolinsky, 2006]) but also

local non-linearities due to membrane saturation. Hence, we can capture fundamental features of dendritic integration by directly deriving the Green's function from dendritic cable theory.

We implemented our synapse model in the Python programming language as a proof of principle, and validated it by evaluating its correctness and execution times on two tasks. First, we show that a morphology-less point-neuron equipped with the proposed synapse model can exploit differential dendritic processing to perform an input-order detection task [Agmon-Snir et al., 1998]. We show that both for passive models and models with active currents in the soma, the agreement with a reference NEURON simulation [Carnevale and Hines, 2006] is seamless. Second, we show that the proposed neuron model is capable of accurate temporal integration of multiple synaptic inputs, a result for which knowledge of the precise neuronal morphology in relation to the synaptic locations is imperative. To this end, we construct a point-neuron model mimicking the dendritic processing in the dendrites of a Layer 5 pyramidal cell. Again, we demonstrate that the agreement with a reference NEURON simulation is seamless. By providing this example, we demonstrate that our proposed approach is highly suitable for the common scenarios to investigate dendritic processing. In such scenarios, the somatic response to a limited number of synapses located in the dendrites is measured while changing the dendritic properties.

2.2 Synapse model based on the Green's function formalism

The core rationale of this work is the simplification of a passive neuron model by analytically computing the transfer function between synapses and the soma. Solving the cable equation for dendrites is not new, and several ways are documented [Koch and Poggio, 1985, Butz and Cowan, 1974, Norman, 1972]. The application of the cable equation to simplify arbitrarily morphologically extended multi-compartmental models to a point-neuron is, however, new.

By solving the cable equation, we thus substitute the effects of an electrical waveform traveling down a dendrite by a so-called pulse-response kernel. Conceptually, we think of the neural response to a spike input as being characterized by three functions: the conductance profile of the synapse, the pulse-response kernel at the synapse and the pulse-response transfer kernel between the input location and the soma to mimic the actual dendritic propagation. The first function is chosen by the modeller: common examples are the alpha function, the double exponential or the single decaying exponential [Rotter and Diesmann, 1999, Giugliano, 2000, Carnevale and Hines, 2006]. The second function captures the decay of the voltage at the synapse given a pulse input, and thus allows for a computation of the synaptic driving force, whereas the third function allows for the computation of the response at the soma, given the synaptic profile, driving force, and dendritic profile.

More formally, we write $g(t)$ for the synaptic conductance profile, $G_{\text{syn}}(t)$ for the pulse response kernel at the synapse and $G_{\text{som}}(t)$ for the pulse response kernel between synapse

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

and soma. Then, given a presynaptic spiketrain $\{t_s\}$ and a synaptic reversal potential E_r , the somatic response of the neuron is characterized by:

$$\begin{aligned} g(t) &= F(\mathbf{a}(t)), \quad \frac{d\mathbf{a}}{dt}(t) = H(\mathbf{a}(t), \{t_s\}) \\ V_{\text{syn}}(t) &= \int_{-\infty}^t dk G_{\text{syn}}(t-k) g(k) (V_{\text{syn}}(k) - E_r) \\ V_{\text{som}}(t) &= \int_{-\infty}^t dk G_{\text{som}}(t-k) g(k) (V_{\text{syn}}(k) - E_r), \end{aligned} \quad (2.1)$$

where E_r is the synaptic reversal potential, $F(\cdot)$ and $H(\cdot)$ depend on the type of synapse chosen and \mathbf{a} denotes the set of synaptic parameters required to generate the conductance profile $g(t)$. Our task is to compute $G_{\text{syn}}(t)$ and $G_{\text{som}}(t)$. We will show that these functions follow from the Green's function formalism.

2.2.1 The neuron model in time and frequency domains

Time domain

Here, we assume a morphological neuron models with passive dendritic segments. Each segment, labeled $d = 1, \dots, N$, is modeled as a passive cylinder of constant radius a_d and length L_d . It is assumed that all segments have an equal membrane conductance g_m , reversal potential E , intracellular axial resistance r_a and membrane capacitance c_m . By convention we label the locations along a dendrite by x , with $x = 0$ and $x = L_d$ denoting the proximal and distal end of the dendrite, respectively. Then, in accordance with cable theory, the voltage in a segment d follows from solving the partial differential equation [Tuckwell, 1988]:

$$\begin{aligned} \frac{\pi a_d^2}{r_a} \frac{\partial^2 V_d}{\partial x^2}(x, t) - 2\pi a_d g_m V_d(x, t) - \\ 2\pi a_d c_m \frac{\partial V_d}{\partial t}(x, t) = I_d(x, t), \end{aligned} \quad (2.2)$$

where $I_d(x, t)$ represents the input current in branch d , at time t and at location x . We assume that the dendritic segments are linked together by boundary conditions that follow from the requirement that the membrane potential is continuous and the longitudinal currents (denoted by I_{ld}) conserved:

$$\begin{aligned} V_d(L_d, t) &= V_i(0, t), \quad i \in \mathcal{C}(d) \\ I_{ld}(L_d, t) &= \sum_{i \in \mathcal{C}(d)} I_{li}(0, t) \end{aligned} \quad (2.3)$$

2.2. Synapse model based on the Green's function formalism

where $\mathcal{C}(d)$ denotes the set of all child segments of segment d . The longitudinal currents are given by:

$$I_{ld}(x, t) = \frac{\pi a_d^2}{r_a} \frac{\partial V_d}{\partial x}(x, t). \quad (2.4)$$

Different dendritic branches originating at the soma are joined together by the lumped-soma boundary condition, which implies for the somatic voltage $V_{\text{som}}(t)$:

$$V_{\text{som}}(t) = V_d(0, t) \quad \forall d \in \mathcal{C}(\text{soma}) \quad (2.5)$$

and

$$\sum_{d=1}^{\mathcal{C}(\text{soma})} I_{ld}(0, t) = I_{\text{som}}(V_{\text{som}}(t)) + C_{\text{som}} \frac{\partial V_{\text{som}}}{\partial t}(t), \quad (2.6)$$

with I_{som} denoting the transmembrane currents in the soma, that can be either passive or active. Note that, for all further calculations, we will treat $I_{\text{som}}(V_{\text{som}}(t))$ as an external input current, and apply the Green's function formalism only on a soma with a capacitive current. For segments that have no children (i.e., the leafs of the tree structure), the sealed end boundary condition is used at the distal end:

$$I_{ld}(L_d, t) = 0 \quad \forall d. \quad (2.7)$$

Frequency domain

Fourier-transforming this system of equations allows for the time-derivatives to be written as complex multiplications, for which analytic [Butz and Cowan, 1974] or semi-analytic [Koch and Poggio, 1985] solutions can be computed. Doing so transforms equation (3.3) into:

$$\frac{\partial^2 V_d}{\partial x^2}(0, \omega) - \gamma_d(\omega)^2 V_d(x, \omega) = I_d(x, \omega) \quad (2.8)$$

where ω is now a complex number and $\gamma_d(\omega)$ is the frequency-dependent space constant, given by

$$\gamma_d(\omega) = \sqrt{\frac{z_{ad}}{z_{md}(\omega)}} \quad (2.9)$$

with $z_{ad} = \frac{r_a}{\pi a_d^2}$ the dendritic axial impedance and $z_{md} = \frac{1}{2\pi a_d(i c_m \omega + g_m)}$ the membrane impedance in branch d . The lumped soma boundary conditions (2.5) and (2.6) become

$$V_{\text{som}}(\omega) = V_d(0, \omega) \quad \forall d \quad (2.10)$$

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

and

$$\sum_{d=1}^N I_{ld}(0, \omega) = \sum_{d=1}^N \frac{1}{z_{ad}} \frac{\partial V_d}{\partial x}(0, \omega) = \frac{1}{Z_{\text{som}}(\omega)} V_{\text{som}}(\omega), \quad (2.11)$$

where

$$Z_{\text{som}}(\omega) = \frac{1}{iC_{\text{som}}\omega} \quad (2.12)$$

is the somatic impedance. The sealed-end boundary conditions are:

$$I_{ld}(L_d, \omega) = \frac{1}{Z_L} V_d(L_d, \omega) = 0 \quad (2.13)$$

with sealed-end impedance $Z_L = \infty$.

2.2.2 Morphological simplification by applying Green's function

Here we will describe the Green's function formalism formally in the time domain to explain the main principles. In the next paragraph we will then turn back to the frequency-domain to compute the actual solution. For the argument we consider a general current input $I_d(x, t)$. In the case of dynamic synapses, such a current input is obtained from the synaptic conductances by the Ohmic relation:

$$I_d(x, t) = g(t)(E_r - V_d(x, t)) \quad (2.14)$$

or, in the case of active channels, from the ion channel dynamics The cable equation (3.3) can be written formally as:

$$\hat{L}_d V_d(x, t) = I_d(x, t) \quad (2.15)$$

where $\hat{L}_d = \frac{\pi a_d^2}{r_a} \frac{\partial^2}{\partial x^2} - 2\pi a_d g_m - 2\pi a_d c_m \frac{\partial}{\partial t}$ is a linear operator¹, which means that for two arbitrary functions $V_1(x, t)$ and $V_2(x, t)$ the following identity holds:

$$\hat{L}_d(aV_1(x, t) + bV_2(x, t)) = a\hat{L}_d V_1(x, t) + b\hat{L}_d V_2(x, t) \quad (2.16)$$

The Green's function of the system is then defined as the solution of the following differential equation:

$$\hat{L}_d G_{dd'}(x, x', t, t') = \delta(x - x')\delta(t - t')\delta_{dd'}. \quad (2.17)$$

¹Note that formally, the operator \hat{L}_d depends on x explicitly in a discontinuous way: for for $0 < x < L_d$: $\hat{L}_d(x) = \frac{\pi a_d^2}{r_a} \frac{\partial^2}{\partial x^2} - 2\pi a_d g_m - 2\pi a_d c_m \frac{\partial}{\partial t}$, and for $x = 0$: $\hat{L}_d(x = 0) = \sum_{d=1}^N \frac{\pi a_d^2}{r_a} \frac{\partial}{\partial x} - G_{\text{som}} - C_{\text{som}} \frac{\partial}{\partial t}$.

2.2. Synapse model based on the Green's function formalism

which also justifies its name as “pulse-response kernel”. The solution to the general input current $I_d(x, t)$ is then written as

$$V_d(x, t) = \sum_{d'} \int_0^{L_d} dx' \int_{-\infty}^t dt' G_{dd'}(x, x', t, t') I_{d'}(x', t'), \quad (2.18)$$

which can be verified by substituting this equation in (2.15) and using the assumption of linearity (2.16).

Two considerations allow us to simplify this system: first, as a consequence of the fact that the operator \hat{L}_d is translation invariant in the time domain, the Green's function only depends on temporal differences:

$$G_{dd'}(x, x', t, t') = G_{dd'}(x, x', t - t'), \quad (2.19)$$

second, in the case of neuronal dynamics, it often suffices to consider inputs at a discrete number of locations, labeled $x_i^{d'}$, with d' denoting the segment of the input location:

$$I_d(x, t) = \sum_{x_i^{d'}} I_d(x, t) \delta(x - x_i^{d'}) \delta_{dd'}, \quad (2.20)$$

where I_d can either denote a synaptic input current or an active membrane current at a point-like location (here we only consider active currents at the soma). Given these considerations, equation (2.18) reduces to

$$V_d(x, t) = \sum_{x_i^{d'}} \int dt' G_{dd'}(x, x_i^{d'}, t - t') I_{d'}(x_i^{d'}, t'). \quad (2.21)$$

Then it follows from equations (2.14) and (2.21) the membrane potentials at the N synapses (labeled i) distributed on dendritic branches d_i at locations $x_1^{d_1}, \dots, x_n^{d_n}$ are

$$\begin{aligned} V_d(x_j^{d_j}, t) = & \sum_i \int_{-\infty}^t dt' G_{d_j d_i}(x_j^{d_j}, x_i^{d_i}, t - t') g_i(t') (E_r - V_{d_i}(x_i^{d_i}, t')) + \\ & \int_{-\infty}^t dt' G_{d_i d_i}(x_j^{d_i}, \text{soma}, t - t') I_{\text{som}}(V_{\text{som}}(t')), \end{aligned} \quad (2.22)$$

whereas the potential at the soma is given by:

$$\begin{aligned} V_{\text{som}}(t) = & \sum_i \int_{-\infty}^t dt' G_{d_i d_i}(\text{soma}, x_i^{d_i}, t - t') g_i(t') (E_r - V_{d_i}(x_i^{d_i}, t')) + \\ & \int_{-\infty}^t dt' G(\text{soma}, \text{soma}, t - t') I_{\text{som}}(V_{\text{som}}(t')). \end{aligned} \quad (2.23)$$

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

Frequency domain solution for the Green's function

Let us now turn to the calculation of the Green's function for a pulse input at time $t = 0$ and at a location x_i in dendrite d . Here, we perform this calculation in the frequency domain, whereas in the next paragraph we will show how the inverse transform, back to the time domain, can be evaluated. To that end we use the algorithm described in [Koch and Poggio, 1985]. As an example, we describe this procedure for a simplified morphology, where each dendritic branch arriving at the soma is modeled as a single cylinder (we use this morphology in section 2.3.3 on input-order detection). In that case the dendrites that do not receive the pulse input merely serve to modify the total somatic impedance. Application of rule I of [Koch and Poggio, 1985] allows us to represent these dendrites (indexed by d') as impedances:

$$Z_{d'}(\omega) = \frac{z_{cd'}(\omega)}{\tanh(\gamma_{d'}(\omega)L_{d'})} \quad (2.24)$$

and rule II allows us to modify the somatic impedance as

$$Z'_{\text{som}}(\omega) = \left(\frac{1}{Z_{\text{som}}(\omega)} + \sum_{d'} \frac{1}{Z_{d'}(\omega)} \right)^{-1} \quad (2.25)$$

Thus the entire effect of the rest of the morphology is summarized in the modified lumped-soma boundary condition

$$I_{ld}(0, \omega) = \frac{1}{Z'_{\text{som}}(\omega)} V_{\text{som}}(\omega). \quad (2.26)$$

The Green's function in the frequency domain at location x then follows from solving equation (2.8) for boundary conditions (2.26) and (2.13), with $I_d(x, t) = \delta(x - x_i)\delta(t)$, and thus $I_d(x, \omega) = \delta(x - x_i)$. From [Butz and Cowan, 1974] it follows that

$$G_{dd}(x, x_i, \omega) = z_{cd}(\omega)^2 \cosh(\gamma_d(\omega)(L_d - x_i)) \cdot \frac{\left(\sinh(\gamma_d(\omega)x) + \frac{Z'_{\text{som}}(\omega)}{z_{cd}(\omega)} \cosh(\gamma_d(\omega)x) \right)}{z_c(\omega) \sinh(\gamma_d(\omega)L_d) + Z'_{\text{som}}(\omega) \sinh(\gamma_d(\omega)L_d)}, \quad (2.27)$$

for $x \leq x_i$, where $z_{cd}(\omega) = \frac{z_{ad}}{\gamma_d(\omega)}$ is the characteristic impedance of dendrite d . Evaluating this function at $x = x_i$ yields the transfer function of synapse i , putting $x = x_j$ ($x_j < x_i$) gives the transfer function between synapse i and synapse j and $x = 0$ results in the transfer function between synapse and soma. The Green's function for $x > x_i$ follows from interchanging x and x_i in (3.80). To compute the effect of a synaptic input on the driving force in other branches (denoted by d'), we first use equation (3.80) (corresponding to rule III of [Koch and Poggio, 1985]) to obtain the pulse-voltage response in the frequency domain at the soma. Then, to compute the pulse voltage response in the branch where the driving force needs to be known, we use

the following identity:

$$G_{dd'}(x, x', \omega) = \frac{G_{d'd'}(x, 0, \omega) G_{dd}(0, x', \omega)}{G(\text{soma}, \text{soma}, \omega)}, \quad (2.28)$$

corresponding to rule IV of [Koch and Poggio, 1985].

Transforming the Green's function to the time domain

Given the conventions we assumed when transforming the original equation, the inverse Fourier transform has following form:

$$G(x, x_i, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega G(x, x_i, \omega) e^{i\omega t}. \quad (2.29)$$

If the Green's function in the time-domain rises continuously from zero, which is generally the case if $x \neq x_i$, it can be approximated with negligible error by the standard technique for evaluating Fourier integrals with the fast-Fourier transform (FFT) algorithm [Press et al., 2007]: we choose a sufficiently large interval $[-\omega_m, \omega_m]$ (where $G(x, x_i, \pm\omega_m)$ is practically 0), divide it in $M = 2^n$ pieces of with $\Delta\omega = \frac{2\omega_m}{M}$ and approximate the integral by a discrete sum:

$$G(x, x_i, t) = \frac{1}{2\pi} \sum_{j=0}^{M-1} G(x, x_i, \omega_j) e^{i\omega_j t}, \quad (2.30)$$

where $\omega_j = -\omega_m + j\Delta\omega$. The choice of discretization step then fixes the timestep $\Delta t = \frac{2\pi}{M\Delta\omega}$. Upon evaluating the Green's function in the time-domain at $t_l = l\Delta t$, $l = 0, \dots, \frac{M}{2} - 1$, expression (2.30) can be written in a form that is suitable for the fast Fourier transform algorithm:

$$G(x, x_i, t_l) = \frac{\Delta\omega}{2\pi} e^{-i\omega_m t_l} \sum_{j=0}^{M-1} G(x, x_i, \omega_j) e^{i\frac{2\pi}{M} j l}, \quad (2.31)$$

and hence:

$$G(x, x_i, t_l) = \frac{M\Delta\omega}{2\pi} e^{-i\omega_m t_l} \text{FFT}(G(x, x_i, \omega_j))_l \quad (2.32)$$

The situation is different if we consider the Green's function at the input location ($x = x_i$). There, the function rises discontinuously from zero at $t = 0$, which causes the spectrum in the frequency-domain to have non-vanishing values at arbitrary high frequencies. Hence, the effect of integrating over a finite interval $[-\omega_m, \omega_m]$ will be non-negligible. Formally, this truncation can be interpreted as multiplying the original function with a window function $H(\omega)$ that is 1 in the interval $[-\omega_m, \omega_m]$ and 0 elsewhere, resulting in a time-domain function

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

that is a convolution of the real function and the transform of the window:

$$\begin{aligned}\tilde{G}(\omega) &= G(x_i, x_i, \omega) H(\omega) \\ \Rightarrow \tilde{G}(t) &= \int_{-\infty}^{\infty} G(x_i, x_i, \tau) H(t - \tau) d\tau.\end{aligned}\tag{2.33}$$

For the rectangular window, the transform $H(t)$ has significant amplitude components for $t \neq 0$, an unwanted property that will cause the Green's function to have spurious oscillations, a phenomenon that is known as spectral leakage [Blackman and Tukey, 1958]. This problem can be solved by choosing a different window function, which is 1 at the center of the spectrum and drops continuously to zero at $-\omega_m$ and ω_m . For this work we found that the Hanning window,

$$H(\omega) = \frac{1}{2} \left(1 + \cos \left(\frac{\pi \omega}{\omega_m} \right) \right),\tag{2.34}$$

gave accurate results for $t \neq 0$. For $t = 0$, the amplitude is slightly underestimated as a consequence of the truncation of the spectrum, whereas for t very close to, but larger than 0, the amplitude is slightly overestimated. However, these errors only cause discrepancy in a very small window (< 0.1 ms) and thus have negligible effect on the neural dynamics.

2.3 Model implementation & Validation

2.3.1 Synapse model implementation

We implemented a prototype of the synapse model discussed above in two stages. First, after specifying the morphology and the synapse locations, the Green's Function is evaluated at the locations that are needed to solve the system, thus yielding a set of pulse response kernels. As modern high-level languages can handle vectorization very efficiently, these functions can be evaluated for a large set of frequencies ω quickly, thus allowing for great accuracy. Second, we implemented a model neuron that uses these Green's functions, sampled at the desired temporal accuracy. Then, given a set of synaptic parameters, the somatic membrane potential is computed by integrating the Volterra-equations (2.22) and (2.23) [Press et al., 2007].

2.3.2 Multi-compartmental and point-neuron model

To compare the performance between a multi-compartmental model and a point-neuron model using the proposed synapse model, we created two comparable neuron models. In the multi-compartmental model, the dendrites are modeled explicitly using NEURON [Carnevale and Hines, 2006], while in the point-neuron model the dendrites are omitted and dendritic processing is carried out implicitly by the new synapse model. The properties of both model neurons are listed in Table 2.1. Evidently, the implicit model has no real morphology and the parameters related to the geometry are used to instantiate the synapse model.

Physiology				
C_m	$1\text{ }\mu F/cm^2$			
g_m	0.02 mS/cm^2			
r_a	$100\text{ }\Omega cm$			
E_l	-65 mV			
Morphology				
Soma length	$25\text{ }\mu m$			
Soma diam	$25\text{ }\mu m$			
	Fig 2.1B		Fig 2.1C	
L_d a_d	dend 1	dend 2	dend 1	dend 2
	$950\text{ }\mu m$	$450\text{ }\mu m$	$900\text{ }\mu m$	$500\text{ }\mu m$
	$0.25\text{ }\mu m$	$0.5\text{ }\mu m$	$0.5\text{ }\mu m$	$1\text{ }\mu m$
Synapses				
E_r τ \bar{g}	syn 1	syn 2	syn 1	syn 2
	0 mV	0 mV	0 mV	0 mV
	1.5 ms	1.5 ms	1.5 ms	1.5 ms
	5 nS	2 nS	20 nS	9 nS

Table 2.1 – **Model neuron parameters.** The multi-compartmental model explicitly simulates the dendritic structure, while the point-neuron is equipped with our model synapse based on Green’s functions and implicitly simulates the dendritic structure. Synaptic time constants were chosen in accordance with canonically accepted values [Roth and van Rossum, 2009].

2.3.3 Input-order detection with differential dendritic filtering

To show the applicability of the new type of model synapse, we use it to perform input-order detection: Suppose a neuron with two dendrites and one synapse (or one group of synapses) on either dendrite (shown in figure 2.1A). In the input-order task, the neuron has to generate a strong response to the temporal activation of the synapses $1 \rightarrow 2$, while generating a weak response to the reversed temporal activation $2 \rightarrow 1$. This behavior is achieved by differential dendritic filtering and can thus not be achieved in a straight-forward way by a single-compartmental model.

We compared the implicit point-neuron model equipped with the new synapse model to the explicit multi-compartmental model in the input-order detection task. The results are illustrated in figure 2.1B. Somatic membrane voltages are shown for the point-neuron model and the multi-compartmental model, after synapse activation in the preferred (left) and null temporal order (right). Because the traces are nearly identical, this result validates our approach and the implementation of the synapse model based on the Green’s function solution to the cable theory.

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

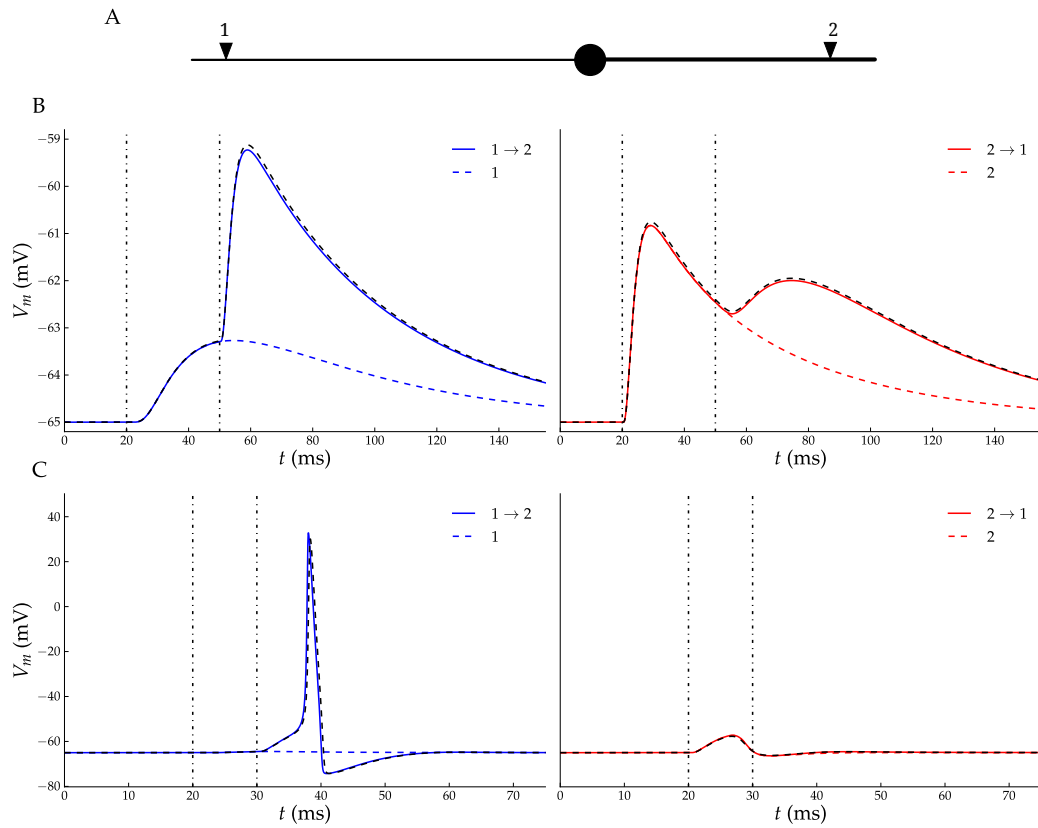


Figure 2.1 – Comparison between a reference multi-compartmental model and a point-neuron model equipped with the new synapse model implicitly simulating dendritic processing. **A:** Both model neurons performed the input-order detection task: The neuron has to respond as strong as possible to the temporal activation $1 \rightarrow 2$ and as weak as possible to the reverse temporal order. **B:** The input-order detection task for a completely passive neuron. Left and right panels contain the somatic membrane potential when the synapses were activated in the preferred ($1 \rightarrow 2$) and null ($2 \rightarrow 1$) temporal order respectively. Colored lines represent the voltage in the point-neuron model and the black dashed line depicts the NEURON trace for comparison. As a reference the waveform when only the first synapse is activated is also shown (left: 1 and right: 2). Vertical dashed-dotted lines denote the spikes arriving at synapse 1 and 2 (left) or 2 and 1 (right). **C** Same as (B), but now the soma contained active HH-currents.

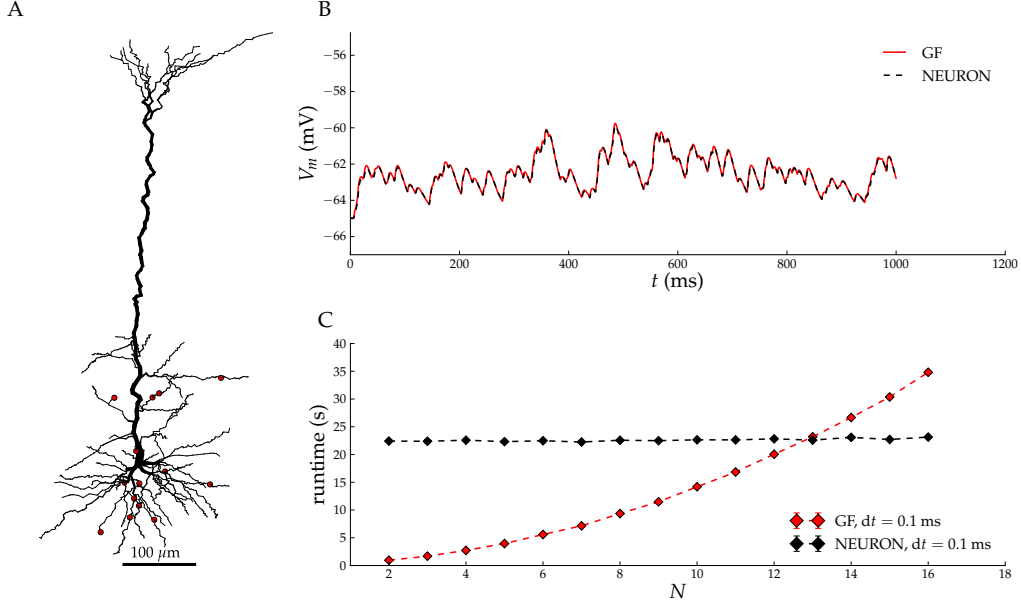


Figure 2.2 – Comparison between the “implicit” (red lines) and “explicit” (black lines) model neurons of a pyramidal cell stimulated by Poisson spiketrains. **A:** The neuron morphology together with the synapse locations. **B:** The membrane potential traces at the soma, for the input locations shown in panel A (red dots). **C:** Comparison of the runtime versus the number of input locations. For few input locations, our prototype python code outperforms the NEURON code.

2.3.4 Voltage-gated active currents

The most prominent non-linear neuronal response is the action potential. Since it is possible in our synapse model to include any non-linear conductance mechanism, as long as it is spatially restricted to a point-like location, we built a prototype containing the Na^+ and K^+ conductances required to generate action potentials. By computing the kernels needed to run the upgraded point-neuron model in the input-order detection task and by adjusting the synaptic weights, we yielded a point-neuron model able to generate a spike in response to the preferred activation pattern, while remaining silent in response to the reversed temporal activation. Note that the active somatic currents shorten the timescale of the neuron’s response compared to the passive model. The timescale of the t -axis was scaled accordingly. In order to validate these outcomes, we again built an equivalent multi-compartmental model in NEURON in which we inserted the same Na^+ and K^+ conductances into the soma. The multi-compartmental model generated identical results, as shown in Figure 2.1C. Thus, in principle we can include conductance descriptions to obtain hallmark neuronal non-linearities.

2.3.5 Multiple synapse interactions

We then checked the correctness of the integrative properties of our implicit point-neuron model by stimulating it with realistic spiketrains at multiple synapses. To that end we added five synapses to a model of a Layer 5 pyramidal neuron equipped with a experimentally reconstructed morphology. The morphology was retrieved from the NeuroMorpho.org repository [Ascoli et al., 2007] and originally published in [Wang et al., 2002]. We stimulated each synapse with Poisson spike trains of rate 10 Hz. The result is shown in Figure 2.2. Again, we compared the implicit model's membrane potential traces to the traces obtained from a multi-compartmental model. The agreement is excellent, as can be seen in Figure 2.2B, which also validates our approach when processing inputs from multiple, interacting synapses.

2.3.6 Runtime

We established that the “implicit” model neuron equipped with our new synapse model generated near-identical voltage traces as a reference multi-compartmental model. Next we compared the run-time of our implementation to the gold standard in multi-compartmental modeling, the NEURON software [Carnevale and Hines, 2006]. To this end we simulated a detailed multi-compartmental model (Figure 2.2) in NEURON as well as with our approach, for increasing numbers of input locations. For each of those numbers we ran three simulations of 1 second of simulated time at an integration step of 0.1 ms (10 kHz). Because in our approach the execution time is independent of the morphological complexity but rather scales with the number of input locations, it is expected that for a low number of input locations, applying our model will be much faster. As shown in Figure 2.2 C, for two input locations, our approach runs 20 times faster than NEURON, while at 13 input locations the execution time is equal. Keeping in mind *i)* that our implementation is done in Python, and *ii)* that often synapses can be grouped together [Pissadaki et al., 2010] we consider this a good outcome.

2.4 Discussion

We presented a bridge between single-compartment and multi-compartmental neuron models by creating a synapse model that analytically computes the dendritic processing between the synaptic input locations and the soma. We then demonstrated that point-neuron models equipped with this new synapse model could flawlessly perform the input-order detection computation; a neuronal computation exploiting differential dendritic processing [Agmon-Snir et al., 1998]. Thus, the new synapse model can be used to introduce computations to point-neurons that previously only belonged to the realm of multi-compartmental neuron models, with a computational cost that does not depend on the morphological complexity.

Then the question arises when it would be advisable to use our synapse model over the standard tools. Although a quantitative comparison should be treated with care due to the different

implementation languages, we still found that our Python-prototype was much faster than the optimized, C++-based NEURON-simulation when the number of input locations was low. This, together with the fact that the computational cost of our model does not depend on morphological complexity, then defines the use case for our model. In scenarios where the number of input locations is low, as is the case in some (invertebrate) cells [Bullock and Horridge, 1965] and as in many *in-silico* scenarios, only few Volterra equations have to be integrated. There our model represents considerable computational advantage. This arguments also holds when more complex neuron types are considered: while cortical neurons receive often as many as 10000 synapses, many of those can be grouped together. To a good approximation, small dendritic branches act as single units, both in terms of short-term input integration [Poirazi et al., 2003b, London and Häusser, 2005] as in terms of long-term plasticity related processes [Govindarajan et al., 2011]. Thus, one could group all synapses in a small branch together and then compute the Green's function for that group of synapses as a hole. Such a grouping would drastically reduce the number of Volterra equations to be integrated and hence enhance performance accordingly.

We assumed that the PSP waveform is transformed only in a passive manner on its way to the soma. In reality, this might sound like a drastic simplification as non-linearity is often cited as a hallmark of neuronal computation, not in the least to generate output spikes. How can we evaluate our synapse model in the light of non-linear computations?

Non-linearities in neural response can occur in two ways. First, at the synapse level a non-linear response can be generated principally through the recruitment of NMDA receptors during repetitive synaptic activation [Branco and Häusser, 2010]. As we assume the evolution in time of the synaptic conductance to be of a known shape, we could -in principle- also mimic a non-linear synaptic conductance by using a more specific description of the synaptic conductance evolution.

Second, non-linearities can arise from voltage-gated conductances in neuronal membranes, that are often distributed non-uniformly along the dendrite [Larkum et al., 1999, Angelo et al., 2007, Mathews et al., 2010]. The distributed nature of voltage-gated conductances leads to the view that dendritic processing is non-linear, and shaped by these conductances and their spatial distributions. Recent work actually challenges this view as it is known that in some behavioral regimes, dendrites act linearly [Ulrich, 2002, Schoen et al., 2012]. Since our Green's function approach relies only on the assumption of linearity, it is not intrinsically restricted to passive dendrites. Ion channels distributed along a dendrite can be linearized [Mauro et al., 1970], and thus yield a quasi-active cable [Koch, 1998]. We anticipate that such a linearization procedure can be plugged into our synapse model, so that the linear (but active) properties of the membrane are captured in the Green's function, yielding accurate and efficient simulations of dendrites that reside in their linear regime. Also, in some cases the actual distribution of voltage-gated conductances along the dendrite does not seem to have any effect as long as the time constant for activation is slower than the spread of voltage itself, which makes the actual location of the voltage-gated conductance irrelevant [Angelo et al., 2007]. Thus, in those cases

Chapter 2. The Green's function formalism as a bridge between single and multi-compartmental modeling

were the spread of voltage is faster than the activation of the conductance, dendrites can act in a passive way, as long as the appropriate non-linearity is introduced at one or a few point-like locations. This can be introduced easily in our synapse model (see Figure 2.1C, with the soma as point-like location with active currents).

While dealing with neuronal non-linearities the focus is often on supra-linear responses to inputs, despite the fact that sub-linear responses are also intrinsically non-linear. Moreover, recently it has been shown both in theory and experiment that sub-linear response are used by neurons [Vervaeke et al., 2012, Abrahamsson et al., 2012]. Even in passive dendrites, sub-linear responses can be generated when the dendrite locally saturates: due to high input resistance the local voltage response to an input can reach the reversal potential of the membrane. At that moment the driving force disappears and a sub-linear response is generated to inputs. This sort of sub-linear response can be generated in conductance-based models with realistic morphologies. Because we implicitly model dendritic morphology, our synapse model is capable of generating these sub-linear responses.

In conclusion, we presented a new synapse model that computes the PSP waveforms as if they were subject to dendritic processing without the need to explicitly simulate the dendrites themselves. With this synapse model comes the ability to simulate dendritic processing at a low computational complexity, that allows it's incorporation in large scale models of neural networks. We thus made a first step to bridge single and multi-compartmental modeling.

Acknowledgements

We thank Marc-Oliver Gewaltig for comments on the manuscript and Moritz Deger for helpful discussion. This work was supported by the BrainScaleS EU FET-proactive FP7 grant.

Author contributions

KMS and BTN designed the research. WW did the simulations and analyzed the results. WW and BTN wrote the paper.

3 A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Willem A.M. Wybo, Daniele Boccalini, Benjamin Torben-Nielsen and Marc-Oliver Gewaltig,
published in *Neural Computation* [Wybo et al., 2015]

Abstract

We prove that when a class of partial differential equations, generalized from the cable equation, is defined on tree graphs, and when the inputs are restricted to a spatially discrete, well chosen set of points, the Green's function (GF) formalism can be rewritten to scale as $O(n)$ with the number n of input locations, contrary to the previously reported $O(n^2)$ scaling. We show that the linear scaling can be combined with an expansion of the remaining kernels as sums of exponentials, to allow efficient simulations of equations from the aforementioned class. We furthermore validate this simulation paradigm on models of nerve cells and explore its relation with more traditional finite difference approaches. Situations in which a gain in computational performance is expected, are discussed.

3.1 Introduction

Neurons have extensive morphological ramifications, called dendrites, that receive and integrate inputs from other neurons, and then transmit the result of this integration to the soma, or cell body, where an output in the form of action potentials is generated. Dendritic integration is considered a hallmark of neuronal computation [London and Häusser, 2005, Häusser and Mel, 2003, Silver, 2010] and parallels the dendritic morphology [Torben-Nielsen and Stiefel, 2010, Agmon-Snir et al., 1998, Segev, 2000]. It is often studied using the cable equation — a one dimensional reaction-diffusion equation that governs the evolution of the membrane potential $V(x, t)$, and is defined on a tree graph representing the dendritic arborization. Inputs, such as

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

synaptic currents, are usually concentrated in a spatially discrete set of points and depend on this potential through their driving force [Kuhn et al., 2004], and possibly through other non-linear conductances such as NMDA [Jahr and Stevens, 1990].

Traditionally, this integration is modeled using compartmental simulations, where space is discretized in a number of compartments of a certain length and a second order finite difference approximation is used to model the longitudinal currents [Hines, 1984]. As the error of this approximation depends on the distance step, complex neural structures require many compartments and are computationally costly to simulate.

Often however, stretches of neural fiber behave approximately linear [Schoen et al., 2012], and one is not interested in the explicit voltage at all locations, but only at a specific output location. For this reason we proposed the idea of using the Green's function formalism to simulate neuron models that receive inputs at a limited number of locations [Wybo et al., 2013]. Given a number of n inputs, the voltage at an output location x_i can be written in this formalism as:

$$V(x_i, t) = \sum_{j=1}^n \int_0^t g(x_i, x_j, t-s) I_j(s) ds. \quad (3.1)$$

Problems arise when these inputs depend on the local voltage. Since, in such a case, all local voltages have to be known, a system of n Volterra integral equations has to be integrated:

$$V(x_i, t) = \sum_{j=1}^n \int_0^t g(x_i, x_j, t-s) I_j(s, V(x_j, s)) ds, \quad i = 1, \dots, n. \quad (3.2)$$

It can be seen that this system contains n^2 convolutions. This unfavorable scaling, along with the fact that the convolutions themselves are costly to compute and the restriction to point-source non-linearities, significantly impedes the computational efficiency of the GF formalism, and restricts its usefulness to very small numbers of input locations [Koch, 1998, page 59-60].

In this work, we are able to significantly improve computational efficiency compared to the classic GF formalism by showing that all three perceived disadvantages can be overcome. Using a transitivity property for the Green's function [Koch, 1998] (see appendix §3.6.1), we show that, when the input locations are well chosen, a transformation of the system (3.2) exists so that only $O(n)$ kernels are required. We term this the *sparse Green's function* (SGF) formalism. As an example of how an efficient integration algorithm can be designed for the resulting system of Volterra integral equations, we show that the kernels can be expressed as sums of exponentials using the vector fitting (VF) algorithm [Gustavsen and Semlyen, 1999] (see appendix §3.6.2) and that consequently the convolutions can be computed recursively. Finally, we show (in a simplified setting) that when the spacing between the input locations becomes sufficiently small, the SGF formalism reduces to the second order finite difference approximation (in the spatial component) of the original equation. As a consequence, the SGF formalism can be seen as a 'generalization' of the second order finite difference approximation

to arbitrary distance steps, as long as what lies in between the distance steps is approximately linear.

We validate this novel SGF formalism by reproducing two canonical results in neuroscientific modeling. First, we reproduce the result of [Moore et al., 1978] on axonal action potential velocity. Second, we compare our SGF formalism with the de facto standard NEURON simulator [Carnevale and Hines, 2006] in the case of dendritic integration with conductance based synapses. In a final section, we discuss in which cases the SGF formalism may yield computational advantage over canonical second order finite difference approaches (of which the NEURON simulator is an example).

3.1.1 The system of equations

Each edge of the tree graph represents a segment of the dendritic tree, for which the cable equation has the following form:

$$2\pi ac_m \frac{\partial V}{\partial t}(x, t) + \frac{\pi a^2}{r_a} \frac{\partial^2 V}{\partial x^2}(x, t) - 2\pi ag_m V(x, t) + \sum_c I_c(x, t) = \sum_{i=1}^n I_i(t, V(x_i, t)) \delta(x - x_i), \quad (3.3)$$

where c_m , g_m and r_a denote, respectively, the membrane capacitance, the membrane conductance and the axial resistance, a denotes the radius of the dendritic branch, I_c the current contribution of a channel type c and I_i the input current at location x_i . The ion channel current can depend non-linearly on the voltage and a number of state-variables:

$$I_c(x, t) = f_c(V(x, t), \mathbf{y}_c(x, t)), \quad (3.4)$$

where the state-variables $\mathbf{y}_c(x, t)$ evolve according to:

$$\dot{y}_{c,j}(x, t) = \frac{y_{c,j,\text{inf}}(V(x, t)) - y_{c,j}(x, t)}{\tau_{c,j}(V(x, t))}, \quad (3.5)$$

with $\tau_{c,j}$ and $y_{c,j,\text{inf}}$ functions that depend on the channel type. Linearizing these currents yields a quasi-active description [Koch, 1998] of the ion channels:

$$I_{c,\text{lin}}(x, t) = \frac{\partial f_c}{\partial V} V(x, t) + \sum_j \frac{\partial f_c}{\partial y_{c,j}} y_{c,j}(x, t), \quad (3.6)$$

with

$$\dot{y}_{c,j}(x, t) = \frac{d}{dV} \left(\frac{y_{c,j,\text{inf}}}{\tau_{c,j}} \right) V(x, t) - \frac{1}{\tau_{c,j}} y_{c,j}(x, t), \quad (3.7)$$

where all derivatives, as well as $\tau_{c,j}$, are evaluated at the equilibrium values of the state variables. If there are a total of K state-variables associated with ion channels, a system of $K + 1$ PDE's of first degree in the temporal coordinate is obtained, which can be recast into a single PDE of degree $K + 1$.

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Consequently, we are interested in the GF of PDE's of the following form:

$$\hat{L}(x)V(x, t) = \left[\hat{L}_0(x) \frac{\partial^2}{\partial x^2} + \hat{L}_1(x) \frac{\partial}{\partial x} + \hat{L}_2(x) \right] V(x, t) = \hat{L}_3(x) \sum_{i=1}^n I_i(t) \delta(x - x_i), \quad (3.8)$$

where $\hat{L}_i(x), i = 0, \dots, 3$ are operators of the form $\hat{L}_i(x) = \sum_{k=0}^{K+1} C_k \frac{\partial^k}{\partial x^k}$, $K \in \mathbb{N}$, and δ is the Dirac-delta function. We assume:

- (i) that an equation of the form (3.8) is defined on each edge of the tree graph (let E denote the set of edges).
- (ii) that on each leaf (let Λ denote the set of leafs) a boundary condition of the following form holds:

$$\hat{L}_1^\lambda \frac{\partial}{\partial x} V^{\epsilon_\lambda}(t) + \hat{L}_2^\lambda V^{\epsilon_\lambda}(t) = \hat{L}_3^\lambda I^\lambda(t) \quad \forall \lambda \in \Lambda, \quad (3.9)$$

where \hat{L}_i^λ are operators defined analogously to $\hat{L}_i(x)$ and where V^{ϵ_λ} is the field value on the adjacent edge ϵ_λ in the limit of x^{ϵ_λ} approaching the leaf. Note that in this general form, equation (3.9) can represent sealed end ($\hat{L}_1^\lambda = 1, \hat{L}_2^\lambda = \hat{L}_3^\lambda = 0$) or voltage clamp ($\hat{L}_1^\lambda = 0, \hat{L}_2^\lambda = \hat{L}_3^\lambda = 1$ and $I(t)$ constant) boundary conditions or, when there is only one neurite leaving the soma, a lumped soma boundary condition (see [Tuckwell, 1988], and where the operators can possibly contain higher order derivatives if quasi-active channels are present).

- (iii) that at each node that is not a leaf (let Φ denote the set of nodes that are not leafs, and let $E(\phi)$ denote the set of edges that join at node $\phi \in \Phi$):

$$\begin{aligned} V^\epsilon(t) &= V^{\epsilon'}(t), \quad \forall \epsilon, \epsilon' \in E(\phi), \quad \forall \phi \in \Phi \\ \sum_{\epsilon \in E(\phi)} \hat{L}^\epsilon \frac{\partial}{\partial x} V^\epsilon(t) &= \hat{L}_1^\phi V^\epsilon(t) + \hat{L}_2^\phi I^\phi(t), \quad \forall \phi \in \Phi, \end{aligned} \quad (3.10)$$

where the operators $\hat{L}^{\epsilon, \phi}$ are again defined as above and V^ϵ denotes the voltage on edge ϵ in the limit of x^ϵ approaching the node ϕ . The first condition then expresses the continuity of the voltage at a node, whereas the second condition can signify the conservation of current flow ($\hat{L}^\epsilon = \pi a_\epsilon^2 / r_a^\epsilon, \hat{L}_1^\phi = \hat{L}_2^\phi = 0$) or a somatic boundary condition when multiple neurites join at the soma (see again [Tuckwell, 1988]).

Algorithms to compute the GF of this system of PDE's have been described extensively in the neuroscientific literature: the algorithm by [Koch and Poggio, 1985] computes the Green's function exactly in the Fourier domain whereas the 'sum-over-trips' approach pioneered by [Abbott et al., 1991] (see [Bressloff and Coombes, 1997] for another overview) and extended by [Coombes et al., 2007, Caudron et al., 2012] uses a path integral formalism. We implemented the algorithm given in [Koch and Poggio, 1985] as we are interested in the GF in the Fourier domain.

3.2 Methods

3.2.1 A sparse reformulation of the Green's function formalism

In this section we prove formally that a sparse reformulation of equation (3.2) exists. Fourier transforming this equation gives the GF formalism in the Fourier domain [Butz and Cowan, 1974, Wybo et al., 2013], :

$$\begin{pmatrix} \tilde{V}_1(\omega) \\ \vdots \\ \tilde{V}_n(\omega) \end{pmatrix} = \begin{pmatrix} g_{11}(\omega) & \cdots & g_{1n}(\omega) \\ \vdots & \ddots & \vdots \\ g_{n1}(\omega) & \cdots & g_{nn}(\omega) \end{pmatrix} \begin{pmatrix} \tilde{I}_1(\omega) \\ \vdots \\ \tilde{I}_n(\omega) \end{pmatrix}, \quad (3.11)$$

where $\tilde{V}_1, \dots, \tilde{V}_n$ denote the field values resulting from the inputs $\tilde{I}_1, \dots, \tilde{I}_n$ at locations $1, \dots, n$ (which may be arbitrarily distributed along the edges of the tree graph), and g_{ij} denotes the GF evaluated between points i and j (see [Koch and Poggio, 1985] for an example of an algorithm to evaluate these functions). Note that for the GF, we dropped the \sim that signifies the Fourier transform for notational simplicity. Whenever the argument of a kernel is not explicitly mentioned, it will be implied that it is the Fourier transform that is under consideration. Note furthermore that the GF between points i and j is equal to the GF between points j and i , so that the matrix in equation (3.11) is symmetric [Koch, 1998, page 63]. Note finally that while the input current \tilde{I}_i may depend on the local voltage (cf. equation (3.3)), it is still possible to take the Fourier transform by considering $I_i(t, V(x_i, t))$ as an a priori unknown function of time ($\equiv I_i(t)$).

We can rewrite the set of equations (3.11) so that the field at one location depends on the input only at that location and the field at all other locations:

$$\tilde{V}_i(\omega) = f_i(\omega) \tilde{I}_i(\omega) + \sum_{j \neq i} h_{ij}(\omega) \tilde{V}_j(\omega), \quad (3.12)$$

where (with G denoting the matrix for which $G_{ij} = g_{ij}$ and G^{-1} its matrix inverse and omitting the argument ω for notational clarity)

$$\begin{aligned} f_i &= 1/(G^{-1})_{ii} \\ h_{ij} &= -(G^{-1})_{ij}/(G^{-1})_{ii}. \end{aligned} \quad (3.13)$$

Intuitively, the field at any location can only depend on the local input and the fields at the neighboring locations. After introducing the necessary definitions and notations, we will prove this intuition formally.

Notation 1. Let A denote a matrix. $A[j; i]$ then denotes the same matrix with the j 'th row and i 'th column deleted.

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

This way, $(-1)^{j+i} \det(A[j; i])$ gives the (j, i) 'th minor of A and the elements of the adjugate matrix of A can be written as:

$$\text{adj}(A)_{ij} = (-1)^{j+i} \det(A[j; i]) \quad (3.14)$$

Then, by using Cramer's rule (see for instance [Horn and Johnson, 2012, pages 17-24]):

$$A^{-1} = \frac{\text{adj}(A)}{\det(A)}, \quad (3.15)$$

equation (3.13) can be written by as:

$$\begin{aligned} f_i &= \det(G) / \det(G[i; i]) \\ h_{ij} &= (-1)^{i+j+1} \det(G[j; i]) / \det(G[i; i]). \end{aligned} \quad (3.16)$$

We specify the discrete set of input locations as follows:

Notation 2. \mathcal{P} denotes a set of n points distributed on a tree graph.

The geometry of the tree graph will induce a nearest neighbor relation on \mathcal{P} . We define this relation in the following way:

Definition 1. Two points $i, j \in \mathcal{P}$ are nearest neighbors if no other point lies on the shortest path between them.

This allows for the definition of sets of nearest neighbors:

Definition 2. A set of nearest neighbors \mathcal{N} is a subset of \mathcal{P} in which each pair of points is a pair of nearest neighbors, and for which no other point can be found in \mathcal{P} that is not in \mathcal{N} but is still a nearest neighbor of every point in \mathcal{N} .

In Fig 3.1A we show an illustration of these sets.

Notation 3. We use m to denote the number of sets of nearest neighbors which can be found in a given \mathcal{P} .

Definition 3. For any set of points $\mathcal{P} = \{1, \dots, n\}$, $G(\mathcal{P})$ is the matrix of transfer kernels:

$$G(\mathcal{P}) = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{pmatrix}, \quad 1, \dots, n \in \mathcal{P} \quad (3.17)$$

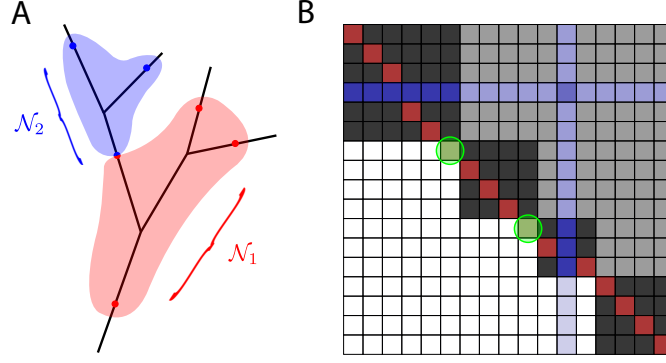


Figure 3.1 – **Schematic of the rationale behind the sparsification of the Green's function formalism for tree structures.** **A:** Illustration of the sets of nearest neighbors \mathcal{N}_q ($q = 1, 2$) induced by the tree structure. For any pair of points in \mathcal{N}_q there is no other point on the shortest path between them. **B:** Illustration of how the structure of the matrices A and $A[j, i]$ (here $j < i$) gives rise to the sparseness of the SGF. These matrices can be written in 'block upper triangular form' (white: zeros, black: blocks on the diagonal, grey: other non-zero elements). The diagonal of $A[j, i]$ (red) is shifted between the j 'th row and i 'th column (blue) compared to the diagonal of A . When j and i do not belong to the same set of nearest neighbors, there is always at least one zero on the diagonal of the resulting matrix (here, there are two zeros — green circles).

However, when \mathcal{P} denotes a set of input locations distributed on the tree graph, we will only make the argument of G explicit when we consider a subset of \mathcal{P} . Hence it is understood that $G \equiv G(\mathcal{P})$, as is the case in formulas (3.13) and (3.16).

Definition 4. An attenuation function a_{ij} is defined by:

$$a_{ij} = g_{ij} / g_{jj}. \quad (3.18)$$

Note that trivially $a_{ii} = 1$.

Definition 5. $A(\mathcal{P})$ is the matrix of attenuation functions a_{ij} between any two points $i, j \in \mathcal{P}$ (analogous to definition 3).

The significance of these sets \mathcal{N} (definition 2) is due to the the following lemma [Koch, 1998, page 63]:

Lemma 1. Transitivity property. If $i, j \in \mathcal{P}$ are not nearest neighbors, then for every point l on the direct path between them it holds that:

$$g_{ij} = \frac{g_{il}g_{lj}}{g_{ll}}, \quad (3.19)$$

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Proof. See appendix §3.6.1. □

This leads directly to a similar transitivity property for attenuations:

$$a_{ij} = a_{il}a_{lj}. \quad (3.20)$$

Note that we may obtain the matrix A (definition 5) from G by dividing, for all $i = 1, \dots, n$, its i 'th column by g_{ii} , and that as a consequence $(\prod_i g_{ii}) \det(A) = \det(G)$. Hence (3.16) can be written as:

$$\begin{aligned} f_i &= g_{ii} \det(A) / \det(A[i; i]) \\ h_{ij} &= (-1)^{i+j+1} \det(A[j; i]) / \det(A[i; i]). \end{aligned} \quad (3.21)$$

Given these definitions and notation, we can capture the aforementioned intuition formally as follows:

Theorem 1. *Consider an arbitrary set of points \mathcal{P} on a tree graph. Then*

i) *for a point i that is in exactly p sets $\mathcal{N}_1, \dots, \mathcal{N}_p$,*

$$f_i = g_{ii} \frac{\det(A(\mathcal{N}_1)) \cdots \det(A(\mathcal{N}_p))}{\det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p)[i; i])} \quad (3.22)$$

ii) *for a pair of points i, j that are nearest neighbors, and for which there consequently exists a set $\mathcal{N} \ni i, j$, and where there are exactly p other sets $\mathcal{N}_1, \dots, \mathcal{N}_p$ that contain i :*

$$h_{ij} = (-1)^{i+j+1} \frac{\det(A(\mathcal{N}_1)) \cdots \det(A(\mathcal{N}_p)) \det(A(\mathcal{N})[j; i])}{\det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p \cup \mathcal{N})[i; i])} \quad (3.23)$$

iii) *for a pair of points i, j that are not nearest neighbors,*

$$h_{ij} = 0 \quad \text{and} \quad h_{ji} = 0. \quad (3.24)$$

Remark 1. To unclutter the notation, we use the indices $i, j \in \mathcal{N}$ to refer to the actual points in \mathcal{P} , their corresponding positions in the full matrices A and in the restricted matrices $A(\mathcal{N})$. This is justified, as it does not influence formulas (3.22) and (3.23). Permuting the numbering of a pair of points amounts to permuting the corresponding rows and columns of the matrices, and thus does not change the determinants. Restricting A to $A(\mathcal{N})$ amounts to deleting the appropriate rows and corresponding columns from A , and thus does not change the factor $(-1)^{i+j+1}$.

Proof. For convenience, we introduce the following ordering scheme for the points: we choose one point as the root of the tree and give it the lowest index, and then reorder the other points such that within each set \mathcal{N} , the point closest to the root has the lowest index, and the other points in \mathcal{N} are numbered consecutively. We then reorder the sets \mathcal{N} so that the set with highest index contains the point with highest index, and so on.

Let there be a total of m sets of nearest neighbors \mathcal{N}_q , $q = 1, \dots, m$ in \mathcal{P} . First, we show that

$$\det(A) = \prod_{q=1}^m \det(A(\mathcal{N}_q)) \quad (3.25)$$

by applying elementary row operations recursively set after set. Let's start with \mathcal{N}_m , the last set of nearest neighbors, containing k elements. Our numbering scheme guarantees that the last $k-1$ rows of A correspond to the $k-1$ elements in \mathcal{N}_m that are not closest to the root. Let us denote the point in \mathcal{N}_m that is closest to the root by l . Between every $d \notin \mathcal{N}_m$ and every $e \in \mathcal{N}_m$ the transitivity property (3.20) holds, and thus $a_{ed} = a_{el}a_{ld}$.

By applying row operations $R_e(A) \rightarrow R_e(A) - a_{el}R_l(A)$, that do not change $\det(A)$ [Horn and Johnson, 2012, pages 9-10], and using (3.20) along with the trivial identity $a_{el} = a_{el}a_{ll}$, the last $k-1$ rows of A become:

$$R_j(A) = \begin{pmatrix} 0 & \dots & 0 & a_{e,n-k+2} - a_{el}a_{l,n-k+2} & \dots & a_{en} - a_{el}a_{ln} \end{pmatrix}. \quad (3.26)$$

Thus, the determinant reduces to

$$\det(A) = \det \begin{pmatrix} A' & A'' \\ 0 & B \end{pmatrix} = \det(A') \det(B), \quad (3.27)$$

where A' and A'' are the parts of A unchanged by the row operations, and B consists of the non-zero elements of the part of A affected by the row operations. It holds that

$$\det(B) = \det(A(\mathcal{N}_m)). \quad (3.28)$$

Construct $A(\mathcal{N}_m)$ by taking the last $k-1$ rows and columns from A , and the attenuations to (from) point l as the first row (column) of $A(\mathcal{N}_m)$. Then, by applying the row-operations $R_e(A(\mathcal{N}_m)) \rightarrow R_e(A(\mathcal{N}_m)) - a_{e1}R_1(A(\mathcal{N}_m))$, the matrix B will be found as the only non-zero minor of the first row elements — the determinant of which is multiplied by 1 to give the determinant of $A(\mathcal{N}_m)$. Thus we have factorized $\det(A)$ in $\det(A') \det(A(\mathcal{N}_m))$.

Our numbering scheme for the points guarantees that a similar reduction can be applied to $\det(A')$, as long as it contains distinct sets \mathcal{N} , which by induction on m proves (3.25). Note that we could have achieved the same reduction by applying the column operations $C_e(A) \rightarrow C_e(A) - a_{le}C_l(A)$ in an analogous manner. Then the matrix in (3.27) would have its zero part in the upper right corner.

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Next we compute the determinant of $A[i; i]$. By removing both the i 'th row and column, it is as if i is removed from the set \mathcal{P} completely. Consequently, $\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p \setminus \{i\}$ forms a new set of nearest neighbors, leading to a factor $\det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p \setminus \{i\})) = \det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p)[i; i])$ instead of $\det(A(\mathcal{N}_1)) \cdots \det(A(\mathcal{N}_p))$ in the final product (3.25). This proves expression (3.22).

Now consider a pair of points i, j as in point *ii*) of the theorem. If either i or j is the point closest in \mathcal{N} to the root, we reorder the points in \mathcal{P} as described above, but now with the extra constraint that within the set \mathcal{N} , all points are numbered consecutively. Note that it is always possible to find such an ordering for any one set \mathcal{N} (but not for all sets at the same time). Assume furthermore that in the ordering of sets, \mathcal{N} comes at the k 'th place ($k \leq m$). Again, we start by factorizing out the determinant of $A(\mathcal{N}_m)$. If $k < m$ and if l , the point in \mathcal{N}_m closest to the root, is not j , the factorization can be carried out as explained above by using the row operations $R_e(A) \rightarrow R_e(A) - a_{el}R_l(A)$ for $e \in \mathcal{N}_m$. If $l = j$, the deletion of the corresponding row prevents us from using it to execute these row operations, but the j 'th column can be used instead: $C_e(A) \rightarrow C_e(A) - a_{le}C_l(A)$. Induction on m , until the set \mathcal{N} is reached, then factors out the determinants of the matrices $A(\mathcal{N}_q)$, $q > k$. If $k = m$, the previous induction can be skipped.

When i nor j are the point closest to the root in set \mathcal{N} , further factorization can proceed unhindered by using row operation, leading to a factorization similar to (3.25), except for the replacement $\det(A(\mathcal{N}_k)) \rightarrow \det(A(\mathcal{N}_k)[j; i])$.

When i (resp. j) is the point closest to the root, our special numbering scheme allows the application of $R_d \rightarrow R_d - d_{di}R_i$ (resp. $C_d \rightarrow C_d - a_{id}C_i$) for $d \in \mathcal{N}_1 \cup \dots \cup \mathcal{N}_{k-1}$, resulting in the matrix:

$$\det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_k)) = \det \begin{pmatrix} B & 0 \\ A'' & A(\mathcal{N})[j; i] \end{pmatrix} = \det(B) \det(A(\mathcal{N})[j; i]), \quad (3.29)$$

resp.

$$\det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_k)) = \det \begin{pmatrix} B & A'' \\ 0 & A(\mathcal{N})[j; i] \end{pmatrix} = \det(B) \det(A(\mathcal{N})[j; i]). \quad (3.30)$$

It can be checked that in both cases $\det(B) = \det(A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_{k-1}))$. Further factorization can then proceed unhindered, leading again to (3.25) with $\det(A(\mathcal{N}_k)) \rightarrow \det(A(\mathcal{N}_k)[j; i])$. This proves expression (3.23).

Finally, consider a pair of points $i > j$ that are not nearest neighbors. We assure that the points are numbered in such a way that all the sets \mathcal{N} that contain j have points with indices smaller than i (note that since i and j are not nearest neighbors, none of these sets contains i). The familiar reduction scheme for $A[j; i]$ by applying row operations can be thought of as writing this matrix in 'block-upper triangular' form. Its determinant is then the product of all determinants of the block-matrices on the diagonal. Between the j 'th row and i 'th column of this matrix, the diagonal is shifted by 1 to a lower row compared to the diagonal of A . Since

i does certainly not belong to the same ‘block’ as j , there is at least one 0 on this diagonal (Fig 3.1B). This proves that $h_{ij} = 0$. That $h_{ji} = 0$, can be proved analogously, but now the matrix is written in ‘block-lower triangular’ form by using column operations. As such, the statement (3.24) is proven. \square

Corollary 1. Number of kernels. *When a total of m sets of nearest neighbors $\mathcal{N}_1, \dots, \mathcal{N}_m$ exists within a set of n points \mathcal{P} , the number of non-zero kernels M in equation (3.12) is:*

$$M = n + \sum_{q=1}^m |\mathcal{N}_q| (|\mathcal{N}_q| - 1), \quad (3.31)$$

with $|\mathcal{N}_q|$ the cardinality of the set \mathcal{N}_q .

Proof. First, we prove equation (3.31). For n points, there are n kernels f_i . In every set \mathcal{N}_q , there is kernels h_{ij} between for every combination of points $i, j \in \mathcal{N}_q$, with $i \neq j$. Consequently, within a set \mathcal{N}_q , there are $|\mathcal{N}_q| (|\mathcal{N}_q| - 1)$ kernels. Hence, in total, there are $n + \sum_{q=1}^m |\mathcal{N}_q| (|\mathcal{N}_q| - 1)$ kernels. \square

Corollary 2. Sparseness. *For a given tree graph and a given positive number n , the configurations of n points that minimize the number M of kernels in (3.12) have:*

$$M = 3n - 2. \quad (3.32)$$

Proof. When $n = 1$ the statement is trivial. For an optimal configuration with $n > 1$, $|\mathcal{N}_q| = 2$ for all m sets \mathcal{N}_q , and $m = n - 1$. Hence $M = n + 2(n - 1) = 3n - 2$. \square

From the viewpoint of computational efficacy, corollaries 1 and 2 indicate that there are ‘well-chosen’ and ‘badly-chosen’ ways for the input locations to be distributed on the tree graph. For a ‘well-chosen’ configuration $|\mathcal{N}| = 2$, or at least $|\mathcal{N}| \ll n$, for all \mathcal{N} , whereas in worst case scenarios there is only a single set \mathcal{N} .

3.2.2 An efficient method to integrate the system of Volterra integral equations

Transforming (3.12) to the time domain results in a system of Volterra integral equations:

$$V_i(t) = \int_0^t f_i(t-s) I_i(s, V(x_i, s)) ds + \sum_{j \neq i} \int_0^t h_{ij}(t-s) V_j(s) ds \quad \forall i, \quad (3.33)$$

where all kernels h_{ij} between points i, j that are not nearest neighbors are zero due to theorem 1. In this form, the SGF-formalism is well-suited to simulate neuron models.

Let $O(n_k)$ denote the number of operations required to compute a convolution each time-step. If these convolutions were to be integrated naively, by evaluating the quadrature explicitly (the Quad approach), n_k would denote the number of time-steps after which the kernel can be

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

truncated. However, the kernels are computed in the frequency domain, so if a partial fraction decomposition in this domain can be found:

$$\begin{aligned} f_i(\omega) &\approx \sum_{l=1}^{L_i} \frac{\gamma_i^l}{i\omega + \alpha_i^l} \\ h_{ij}(\omega) &\approx \sum_{l=1}^{L_{ij}} \frac{\gamma_{ij}^l}{i\omega + \alpha_{ij}^l}, \end{aligned} \quad (3.34)$$

where $\Re(\alpha) < 0$, then in the time domain the kernels can be readily expressed as sums of one sided exponentials:

$$\begin{aligned} f_i(t) &\approx \sum_{l=1}^{L_i} \gamma_i^l e^{\alpha_i^l t} \\ h_{ij}(t) &\approx \sum_{l=1}^{L_{ij}} \gamma_{ij}^l e^{\alpha_{ij}^l t}, \end{aligned} \quad (3.35)$$

when $t \geq 0$, and $f_i(t) = h_{ij}(t) = 0$ otherwise. Such a decomposition can be derived accurately by employing the vector fitting (VF) algorithm [Gustavsen and Semlyen, 1999] (see appendix §3.6.2). Consequently, the convolutions with the kernels become sums of convolutions with exponentials, which can be readily expressed as simple differential equation (see for instance [Lubich and Schädle, 2002]). We call this the Exp approach and n_k is the number of exponentials per kernel in this case. Usually, for the Exp approach, n_k is smaller than for the Quad approach. Nevertheless, many of the exponentials of the VF algorithm are used to approximate the small t behavior of the kernels, and hence have a very short time-scale. The large t behaviour is often described by one or a few exponentials. This suggest that a ‘mixed’ approach could yield optimal performance, where for small t the quadrature is computed explicitly, and for large t the convolution is computed as an ODE (or the sum of a few ODE’s). We now give a detailed account of the mixed approach.

Let $t = Nh$, with h the integration step and N a natural number, and let us assume that $V_i(\tau)$ is known for all i and for $\tau \in \{t - kh; k = 1, \dots, N\}$, and that we want to know $V_i(t + h)$, with $h > 0$. We split the convolutions in equation (3.33) into an quadrature term (\int_{t-Kh}^{t+h}) and a exponential term (\int_0^{t-Kh}):

$$\begin{aligned} V_i(t + h) &= \int_{t-Kh}^{t+h} f_i(t + h - s) I_i(s) ds + \sum_{j \neq i} \int_{t-Kh}^{t+h} h_{ij}(t + h - s) V_j(s) ds + \\ &\quad \int_0^{t-Kh} f_i(t + h - s) I_i(s) ds + \sum_{j \neq i} \int_0^{t-Kh} h_{ij}(t + h - s) V_j(s) ds, \end{aligned} \quad (3.36)$$

where K is a natural number that needs to be chosen. We assume that between the temporal

grid points $t - kh$ and $t - (k + 1)h$, $V_j(t)$ can be approximated by a linear interpolation:

$$V_j(s) \approx V_j(t - kh) + \frac{V_j(t - kh) - V_j(t - (k + 1)h)}{h}(s - t + kh), \quad t - (k + 1)h \leq s \leq t - kh. \quad (3.37)$$

With this and the exponential approximation (3.35), the quadrature term for convolutions with V_j becomes:

$$\begin{aligned} \int_{t-Kh}^{t+h} h_{ij}(t+h-s)V_j(s)ds \approx & V_j(t+h) \sum_l \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^l} + \frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} (e^{\alpha_{ij}^l h} - 1) \right] \\ & + \sum_{k=0}^{K-1} V_j(t-kh) \sum_l \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} (e^{\alpha_{ij}^l (k+1)h} - 2e^{\alpha_{ij}^l kh} + e^{\alpha_{ij}^l (k-1)h}) \right] \\ & + V_j(t-Kh) \sum_l \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^l} e^{\alpha_{ij}^l Kh} + \frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} (e^{\alpha_{ij}^l Kh} - e^{\alpha_{ij}^l (K-1)h}) \right], \end{aligned} \quad (3.38)$$

and similarly for the convolutions with the input I_i . For the exponential term, we get the following:

$$\int_0^{t-Kh} h_{ij}(t+h-s)V_j(s)ds = \sum_l \gamma_{ij}^l e^{\alpha_{ij}^l (K+1)h} \int_0^{t-Kh} e^{\alpha_{ij}^l (t-Kh-s)} V_j(s)ds, \quad (3.39)$$

where $u_{ij}^l(t-Kh) \equiv \int_0^{t-Kh} \gamma_{ij}^l e^{\alpha_{ij}^l (t-Kh-s)} V_j(s)ds$ is the solution of an initial value problem:

$$\begin{aligned} \dot{u}_{ij}^l(t) &= \alpha_{ij}^l u_{ij}^l(t) + \gamma_{ij}^l V_j(t) \\ u_{ij}^l(t=0) &= 0, \end{aligned} \quad (3.40)$$

whose value can be computed recursively:

$$u_{ij}^l(t-Kh) = e^{\alpha_{ij}^l h} u_{ij}^l(t-(K+1)h) + \int_{t-(K+1)h}^{t-Kh} e^{\alpha_{ij}^l s} V_j(s)ds. \quad (3.41)$$

Using again the linear approximation (3.37), this becomes:

$$\begin{aligned} u_{ij}^l(t-Kh) \approx & e^{\alpha_{ij}^l h} u_{ij}^l(t-(K+1)h) \\ & + V_j(t-Kh) \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^l} + \frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} (e^{\alpha_{ij}^l h} - 1) \right] \\ & + V_j(t-(K+1)h) \left[\frac{\gamma_{ij}^l}{\alpha_{ij}^l} e^{\alpha_{ij}^l h} - \frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} (e^{\alpha_{ij}^l h} - 1) \right]. \end{aligned} \quad (3.42)$$

Analogous considerations apply for the convolutions with the input I_i .

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Let us now define a matrix H_0 by:

$$(H_0)_{ij} = \sum_l \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^l} + \frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} (e^{\alpha_{ij}^l h} - 1) \right] \quad (3.43)$$

and a tensor H_1 by:

$$\begin{aligned} (H_1)_{ij}^k &= \sum_l \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} \left(e^{\alpha_{ij}^l (k+1)h} - 2e^{\alpha_{ij}^l kh} + e^{\alpha_{ij}^l (k-1)h} \right) \right] \quad \text{for } k = 0, \dots, K-1 \\ (H_1)_{ij}^k &= \sum_l \left[-\frac{\gamma_{ij}^l}{\alpha_{ij}^l} e^{\alpha_{ij}^l kh} + \frac{\gamma_{ij}^l}{\alpha_{ij}^{l2}h} \left(e^{\alpha_{ij}^l kh} - e^{\alpha_{ij}^l (K-1)h} \right) \right] \quad \text{for } k = K. \end{aligned} \quad (3.44)$$

when i, j are nearest neighbors, and $(H_0)_{ij} = (H_1)_{ij}^k = 0$ otherwise. For the input kernels f_i , we define a vector F_0 (with $(F_0)_i$ analogous to (3.43)) and a matrix F_1 (with $(F_1)_i^k$ analogous to (3.44)). Using these definitions, we may write equation (3.36) as:

$$\begin{aligned} V_i(t+h) &= \left[(F_0)_i I_i(t+h) + \sum_{k=0}^K (F_1)_i^k I_i(t-kh) \right] \\ &+ \sum_{j \neq i} \left[(H_0)_{ij} V_j(t+h) + \sum_{k=0}^K (H_1)_{ij}^k V_j(t-kh) \right] \\ &+ \sum_l e^{\alpha_i^{l(K+1)}h} u_i^l(t-Kh) + \sum_{j \neq i} \sum_l e^{\alpha_{ij}^{l(K+1)}h} u_{ij}^l(t-Kh). \end{aligned} \quad (3.45)$$

To simplify the notation, we group all terms that do not contain voltage or input at time $t+h$ in a vector $\mathbf{k}(t)$, for which:

$$\begin{aligned} k_i(t) &= \left[\sum_{k=0}^K (F_1)_i^k I_i(t-kh) \right] + \sum_{j \neq i} \left[\sum_{k=0}^K (H_1)_{ij}^k V_j(t-kh) \right] \\ &+ \sum_l e^{\alpha_i^{l(K+1)}h} u_i^l(t-Kh) + \sum_{j \neq i} \sum_l e^{\alpha_{ij}^{l(K+1)}h} u_{ij}^l(t-Kh) \end{aligned} \quad (3.46)$$

Consequently, equation (3.45) becomes in matrix form:

$$(\mathbb{I} - H_0)\mathbf{V}(t+h) = \text{diag}(\mathbf{F}_0)\mathbf{I}(t+h) + \mathbf{k}(t). \quad (3.47)$$

Solving this matrix equation then gives $\mathbf{V}(t+h)$.

Note that when the input current I_i depends on the local voltage, equation (3.47) is only semi-implicit, since the voltage at time t would be required to compute the current at time $t+h$. This description may be unstable for certain ion channels. However, the direct dependence of most currents on the voltage is linear (in the case of ion-channels for instance, non-linearities

arise through the non-linear dependence of state variables on the voltage):

$$I_i(t, V(x_i, t)) = c_i(t) + d_i(t)V(x_i, t). \quad (3.48)$$

Using this, equation (3.47) can be modified in the following way:

$$(\mathbb{I} - H_0 - \text{diag}(\mathbf{F}_0 \odot \mathbf{d}(t+h))\mathbf{V}(t+h) = \text{diag}(\mathbf{F}_0)\mathbf{c}(t+h) + \mathbf{k}(t). \quad (3.49)$$

where \odot denotes the element wise multiplication. This way of integration, while still being semi-implicit (since the equations for state variables of ion channels (3.5) are still integrated explicitly) is stable and standard in neuroscientific applications [Hines, 1984].

Note furthermore that the matrix $(\mathbb{I} - H_0)$ resp. $\mathbb{I} - H_0 - \text{diag}(\mathbf{F}_0 \odot \mathbf{d}(t+h))$ is structurally symmetric: whenever an off-diagonal element is nonzero, its counterpart opposite to the diagonal is nonzero as well. When $|\mathcal{N}| = 2$ for all \mathcal{N} , and when the input locations are ordered in the right way, this matrix is a Hines matrix [Hines, 1984]. For such a matrix a linear system of equations of the form (3.47) resp. (3.49) can be solved for \mathbf{V} in $O(n)$ steps instead of the usual $O(n^3)$ steps.

Note also that we did not discuss the initialization steps of this algorithm explicitly. We omitted this discussion since, in neuroscience, it usually is simply assumed that the neuron model is at equilibrium at all times $t < 0$. Hence, the vectors $\mathbf{k}(t=0), \mathbf{k}(t=h), \mathbf{k}(t=2h), \dots$ can be computed easily by assuming that $V(-kh) = 0$ for all required k 's.

Finally, we remark that K , the parameter in the algorithm which determines the limit Kh below which the quadrature is evaluated explicitly, can be chosen to minimize the number n_k of operations per kernel. In principle, K could be chosen separately for each kernel. In the present derivation we however opted not to do so for simplicity.

3.2.3 Nonlinear terms and the small Δx limit of the SGF formalism

While a set of ion channels that behaves approximately linear may be incorporated directly in the SGF formalism, the question what to do with channels that act truly non-linear remains. Such channels can be moved from the left hand side of equation (3.3) to the right hand side, and treated as an input current that depends on the local voltage. This current needs to be 'compartmentalized', i.e. expressed at a discrete number of input locations, with a certain separation Δx .

A 'recompartmentalization' of this type leads to the question of what the relation between the SGF formalism and the canonical second order finite difference approximation is. In both approaches, the input is compartmentalized. In the finite difference approximation however, the entire cable (i.e. the longitudinal, capacitive and leak currents) is compartmentalized as well, whereas in the SGF formalism, the cable is treated exactly. Consequently, in spatial regions with truly non-linear ion channels, the accuracy of the SGF is equivalent to (or better

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

than, since the linear currents are still treated exactly) the accuracy of the canonical second order finite difference approximation.

The observation that the SGF formalism treats the cable exactly then leads to the question whether the second order finite difference approximation can be derived from it, as both approaches describe the voltage at a given location only as a function of the voltage at the neighboring locations and the input at that location. We show in a simplified setting that, when the distance Δx between the input locations is sufficiently small, the formalism reduces to the second order finite difference approximation of the original equation.

Although we expect that the reduction of the SGF formalism to the second order finite difference approximation is valid for all equations of the type (3.8) and for arbitrary tree graphs, its derivation can become prohibitively complex. We therefore constrain ourselves to the passive cable equation (3.3) and to the case where this equation is defined on a line of length 1, with a homogeneous boundary condition at each end. We suppose that there are $n - 1$ input locations, distributed evenly with spacing $\Delta x = 1/n$.

$$\begin{cases} \frac{\partial V}{\partial t}(x, t) + \frac{\partial^2 V}{\partial x^2}(x, t) - V(x, t) = \sum_{i=1}^{n-1} I_i(t) \delta(x - i\Delta x). \\ \frac{\partial V}{\partial x}(0, t) = \hat{L}_A V(0, t) \\ \frac{\partial V}{\partial x}(1, t) = \hat{L}_B V(1, t) \end{cases} \quad (3.50)$$

Note that the input currents $I_i(t)$ can depend on the local voltage $V(x_i, t)$, as they can be the result of a discretization of the ion channel currents.

Second order finite difference approximation To obtain the second order finite difference approximation of equation (3.50), we replace

$$\frac{\partial^2 V}{\partial x^2}(x_i, t) \rightarrow \frac{V_{i+1}(t) - 2V_i(t) + V_{i-1}(t)}{\Delta x^2} \quad (3.51)$$

and we average the inputs for each compartment:

$$\frac{1}{\Delta x} \int_{(i-1/2)\Delta x}^{(i+1/2)\Delta x} I_i(t) \delta(x - i\Delta x) dx = \frac{I_i(t)}{\Delta x} \quad (3.52)$$

Consequently, we find for $i = 1, \dots, n - 1$:

$$\frac{V_{i+1}(t) - 2V_i(t) + V_{i-1}(t)}{\Delta x^2} + \dot{V}_i(t) - V_i(t) = \frac{I_i(t)}{\Delta x}. \quad (3.53)$$

The boundary condition at $x = 0$ becomes:

$$\frac{V_1(t) - V_0(t)}{\Delta x} = \hat{L}_A^{(t)} V_0(t), \quad (3.54)$$

and an analogous expression applies for the boundary condition at $x = 1$

Reduction of the SGF formalism The Fourier transform of the problem (3.50) reads:

$$\begin{cases} \frac{\partial^2 \tilde{V}}{\partial x^2}(x, \omega) - \gamma(\omega)^2 \tilde{V}(\omega) = \sum_{i=1}^n \tilde{I}_i(\omega) \delta(x - x_i) \\ \frac{\partial \tilde{V}}{\partial x}(0, \omega) = \tilde{L}_A(\omega) \tilde{V}(0, \omega) \\ \frac{\partial \tilde{V}}{\partial x}(1, \omega) = \tilde{L}_B(\omega) \tilde{V}(1, \omega), \end{cases} \quad (3.55)$$

where \tilde{L}_A and \tilde{L}_B are polynomials in ω representing the respective Fourier transforms of \hat{L}_A and \hat{L}_B , and where $\gamma^2(\omega) = i\omega - 1$. In the following the coordinate ω is dropped for notational clarity. We define two linearly independent solutions to the homogeneous problem:

$$\begin{aligned} u_A(x) &= e^{\gamma x} + k_A e^{-\gamma x} \\ u_B(x) &= e^{\gamma x} + k_B e^{-\gamma x} \end{aligned} \quad (3.56)$$

that satisfy the boundary conditions respectively in $x = 0$ (u_A) and $x = 1$ (u_B). Then the Green's function of problem (3.55) is given by [Stakgold, 1967, page 66]:

$$g_{ij} = \begin{cases} \frac{u_A(x_i)u_B(x_j)}{-2\gamma(k_B - k_A)} & \text{if } x_i \leq x_j \\ \frac{u_B(x_i)u_A(x_j)}{-2\gamma(k_B - k_A)} & \text{if } x_i \geq x_j. \end{cases} \quad (3.57)$$

Consequently, the attenuation functions are given by:

$$a_{ij} = \begin{cases} \frac{u_A(x_i)}{u_A(x_j)} & \text{if } x_i \leq x_j \\ \frac{u_B(x_i)}{u_B(x_j)} & \text{if } x_i \geq x_j. \end{cases} \quad (3.58)$$

According to theorem 1, problem (3.55) is then fully defined by the following kernels:

- for $i = 0, \dots, n-2$

$$h_{i+1,i} = \frac{a_{i+1,i} - a_{i+1,i+2}a_{i+2,i}}{1 - a_{i,i+2}a_{i+2,i}} = a_{i+1,i} \frac{1 - a_{i+1,i+2}a_{i+2,i+1}}{1 - a_{i,i+2}a_{i+2,i}} \quad (3.59)$$

- for $i = 2, \dots, n-1$

$$h_{i,i+1} = \frac{a_{i,i+1} - a_{i,i-1}a_{i-1,i+1}}{1 - a_{i+1,i-1}a_{i-1,i+1}} = a_{i,i+1} \frac{1 - a_{i,i-1}a_{i-1,i}}{1 - a_{i+1,i-1}a_{i-1,i+1}} \quad (3.60)$$

- $h_{01} = a_{01}$

- $h_{n,n-1} = a_{n,n-1}$

- for $i = 1, \dots, n-1$

$$f_i = g_{ii} \frac{(1 - a_{i,i-1}a_{i-1,i})(1 - a_{i+1,i}a_{i,i+1})}{1 - a_{i+1,i-1}a_{i-1,i+1}} \quad (3.61)$$

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Using equations (3.58) and (3.56), it can be checked that:

$$1 - a_{ij}a_{ji} = \begin{cases} \frac{2(k_B - k_A) \sinh(\gamma(x_j - x_i))}{u_B(x_i)u_A(x_j)} & \text{if } x_i \leq x_j \\ \frac{2(k_B - k_A) \sinh(\gamma(x_i - x_j))}{u_A(x_i)u_B(x_j)} & \text{if } x_i \geq x_j. \end{cases} \quad (3.62)$$

Consequently, it follows that:

- for $i = 0, \dots, n-2$

$$h_{i+1,i} = \frac{\sinh(\gamma\Delta x)}{\sinh(2\gamma\Delta x)} \quad (3.63)$$

- for $i = 2, \dots, n-1$

$$h_{i,i+1} = \frac{\sinh(\gamma\Delta x)}{\sinh(2\gamma\Delta x)} \quad (3.64)$$

- for $i = 1, \dots, n-1$

$$f_i = -\frac{1}{\gamma} \frac{(\sinh(\gamma\Delta x))^2}{\sinh(2\gamma\Delta x)} \quad (3.65)$$

With these equations, equation (3.12) for $i = 1, \dots, n-1$ becomes:

$$\tilde{V}_i = -\frac{1}{\gamma} \frac{(\sinh(\gamma\Delta x))^2}{\sinh(2\gamma\Delta x)} \tilde{I}_i + \frac{\sinh(\gamma\Delta x)}{\sinh(2\gamma\Delta x)} (\tilde{V}_{i-1} + \tilde{V}_{i+1}). \quad (3.66)$$

Since Δx is small, the sinh-functions can be expanded. This gives:

$$\tilde{V}_i = -\frac{\Delta x}{2 + \frac{8}{6}\gamma^2\Delta x^2} \tilde{I}_i + \frac{1 + \frac{1}{6}\gamma^2\Delta x^2}{2 + \frac{8}{6}\gamma^2\Delta x^2} (\tilde{V}_{i-1} + \tilde{V}_{i+1}). \quad (3.67)$$

Multiplying both sides by the denominator $2 + \frac{8}{6}\gamma^2\Delta x^2$, rearranging terms and dividing by Δx^2 then gives:

$$\frac{\tilde{V}_{i-1} - 2\tilde{V}_i + \tilde{V}_{i+1}}{\Delta x^2} - \frac{8}{6}\gamma^2\tilde{V}_i + \frac{1}{6}\gamma^2(\tilde{V}_{i-1} + \tilde{V}_{i+1}) = \frac{1}{\Delta x} \tilde{I}_i. \quad (3.68)$$

Averaging $\tilde{V}_{i-1} + \tilde{V}_{i+1} \approx 2\tilde{V}_i$ and transforming the resulting equation back to the time domain then results precisely in the finite difference approximation for $i = 1, \dots, n-1$.

Let us now investigate equation (3.12) when $i = 0$. Here, it holds that:

$$\tilde{V}_0 = a_{01} \tilde{V}_1. \quad (3.69)$$

Substituting the explicit form of $a_{01} = u_A(0)/u_A(\Delta x)$ in this equation and expanding the exponentials up to first order leads to:

$$\tilde{V}_0 = \frac{1}{1 + \frac{1-k_A}{1+k_A}\gamma\Delta x} \tilde{V}_1 \quad (3.70)$$

Since u_A satisfies the boundary condition at $x = 0$, it can be checked that $\gamma(1-k_A)/(1+k_A) = \tilde{L}_A$, so that equation (3.70) can be rewritten as:

$$\tilde{L}_A \tilde{V}_0 = \frac{\tilde{V}_1 - \tilde{V}_0}{\Delta x}. \quad (3.71)$$

Transforming this equation back to the time domain yields precisely the finite difference approximation for V_0 . Analogous considerations apply for V_n .

3.2.4 Implementation

In this section we summarize the steps one has to undertake to implement the SGF formalism. Such an implementation consists of two main parts: an initialization part and a simulation part.

The initialization part consists of the following steps:

1. For any given set of input locations and a tree graph, a routine has to be implemented that can identify the sets of nearest neighbors. Such a routine could for instance return a list, where each element is a list in itself, containing the input locations that constitute a set of nearest neighbors.
2. The GF's $g_{ij}(\omega)$ between every pair of elements in the sets of nearest neighbors have to be computed. We implemented the algorithm of [Koch and Poggio, 1985] for this purpose. Note that for a set \mathcal{N} of cardinality $|\mathcal{N}|$, only $|\mathcal{N}|(|\mathcal{N}|+1)/2$ function g_{ij} have to be computed, since $g_{ij} = g_{ji}$ [Koch, 1998]. From these kernels, the attenuation functions can be derived easily.
3. The kernels f_i and h_{ij} have to be computed. This can be done either by evaluating the formulas (3.22) and (3.23) explicitly (when all matrices of which the determinants have to be computed are small) or by the derived formulas:

$$\begin{aligned} f_i &= g_{ii} \frac{1}{A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p)^{-1}} \\ h_{ij} &= - \frac{A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p \cup \mathcal{N})_{ij}^{-1}}{A(\mathcal{N}_1 \cup \dots \cup \mathcal{N}_p \cup \mathcal{N})_{ii}^{-1}}. \end{aligned} \quad (3.72)$$

Note that in both cases, attenuation function are needed that are not directly computed in the previous step, since attenuation matrices of unions of sets are considered. This

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

is not a problem, as these attenuation functions can be easily reconstructed from the transitivity property.

4. The partial fraction decomposition of each kernel has to be computed. The VF algorithm [Gustavsen and Semlyen, 1999] (appendix §3.6.2) is well suited for this purpose. When the parameters of the partial fraction decomposition are known, the vector F_0 , the matrices F_1 and H_0 and the tensor H_1 can be computed, as described in (3.43) and (3.44). Since H_0 resp. H_1 are sparse in their indices i and j , they can be stored as index-number resp. index-array pairs.

For the simulation part, two main routines have to be called each time step:

1. A routine that computes the vectors $\mathbf{c}(t)$ and $\mathbf{d}(t)$ (3.48). This routine also advances synaptic conductances and ion channel state variables and its details are well established in the neuroscientific literature (see [Rotter and Diesmann, 1999] and [Hines, 1984]).
2. A routine that first computes the vector $\mathbf{k}(t)$ (3.46) and then solves equation (3.49).

We implemented a prototype of the above outlined initialization algorithm in Python. Tree structures were implemented using the btmorph library [Torben-Nielsen, 2014]. The simulation algorithm was implemented both in pure Python and in C++ with a Cython interface.

3.3 Validation

In myelinated axons, the approximation of grouping active currents at a discrete set of input locations holds exactly, as the only spots where active currents are present are the nodes of Ranvier. These nodes are separated by stretches of myelinated fiber of up to 2 mm in length, which can be modeled by equation (3.3). We reproduce the model of [Moore et al., 1978], where the soma, axon initial segment and nodes of Ranvier are equipped with Hodgkin-Huxley [Hodgkin and Huxley, 1952] channels, so that the input current is of the form:

$$I_i(t) = -\bar{g}_{Na}m_i(t)^3h_i(t)(V(x_i, t) - E_{Na}) + \bar{g}_Kn_i(t)^4(V(x_i, t) - E_K) - \bar{g}_L(V(x_i, t) - E_L). \quad (3.73)$$

Consequently, $c_i(t)$ and $d_i(t)$ in equation (3.48) are given by:

$$\begin{aligned} c_i(t) &= \bar{g}_{Na}m_i(t)^3h_i(t)E_{Na} + \bar{g}_Kn_i(t)^4E_K + \bar{g}_LE_L \\ d_i(t) &= -\bar{g}_{Na}m_i(t)^3h_i(t) - \bar{g}_Kn_i(t)^4 - \bar{g}_L, \end{aligned} \quad (3.74)$$

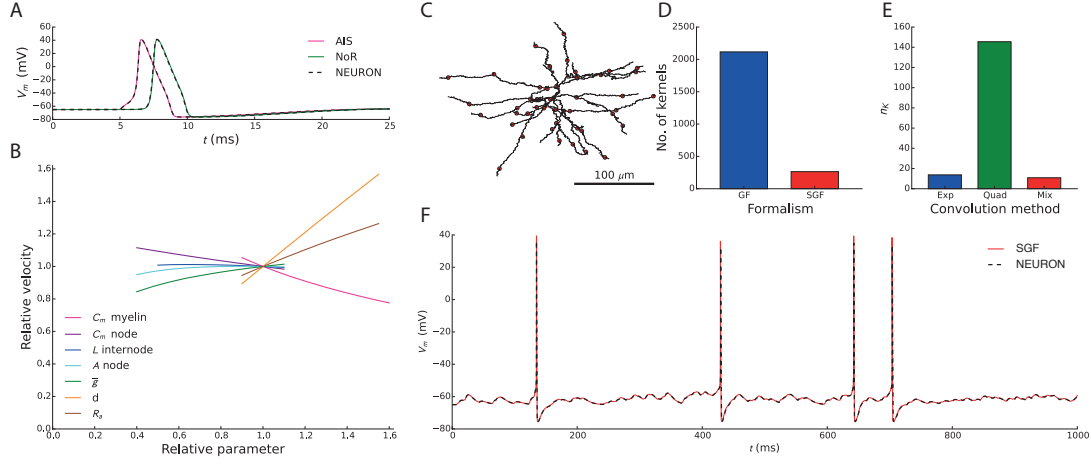


Figure 3.2 – Validation of the SGF formalism on axon (A-B) and dendrite models (C-F). We implemented the axon model described in [Moore et al., 1978] in the SGF formalism. **A:** validation of the action potential (AP) shape by comparison with an equivalent NEURON model at the axon initial segment (AIS) and the 10'th node of Ranvier (NoR). **B:** reproduction of Fig 2 in [Moore et al., 1978], where the dependence of AP velocity on different biophysical parameters is studied. **C:** The dendritic morphology together with 50 synaptic input locations (the morphology was retrieved from the NeuroMorpho.org repository [Ascoli, 2006] and originally published in [Wang et al., 2002]). **D:** The number of kernels in the SGF formalism, compared to the number of kernels that would have been required in the normal GF formalism. **E:** The number of operation required per kernel to achieve similar levels of accuracy for 3 approaches: Exp, where all the exponentials from the VF algorithm are integrated, Quad, where the quadrature is computed explicitly and Mix, where we compute the quadrature for the first K (here 3) steps, and use the ODEs to compute the rest of the convolution. **F:** Voltage trace at the soma upon stimulation of the synapses with 5 Hz Poisson spike trains.

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

and the variables $m_i(t)$, $h_i(t)$ and $n_i(t)$ evolve according to the following equations:

$$\begin{aligned}\dot{m}_i(t) &= \frac{m_{\text{inf}}(V(x_i, t)) - m_i(t)}{\tau_m(V(x_i, t))} \\ \dot{h}_i(t) &= \frac{h_{\text{inf}}(V(x_i, t)) - h_i(t)}{\tau_h(V(x_i, t))} \\ \dot{n}_i(t) &= \frac{n_{\text{inf}}(V(x_i, t)) - n_i(t)}{\tau_n(V(x_i, t))},\end{aligned}\tag{3.75}$$

where m_{inf} , h_{inf} , n_{inf} , τ_m , τ_h and τ_n are functions of the local voltage.

We implemented this model using the SGF formalism. In Fig 3.2A, we validate the action potential (AP) shape at the axon initial segment and at the 10'th node of Ranvier by comparison with the NEURON simulator [Carnevale and Hines, 2006] – the gold standard in neuronal modeling. In Fig 3.2B we reproduce Fig 2 in [Moore et al., 1978], where the dependence of the AP velocity on various axonal parameters is investigated.

Finally, to show that the SGF formalism can also handle complex trees, we equipped a stellate cell morphology, comprising an active soma and passive dendrites (Fig 3.2C) (modeled by equation (3.3)), with 50 conductance based synapses, so that

$$\begin{aligned}I_i(t) &= \bar{g}(A_i(t) - B_i(t))(V(x_i, t) - E_r) \\ \dot{A}_i(t) &= \frac{-A_i(t) + c_A \sum_j \delta(t - s_i^{(j)})}{\tau_A} \\ \dot{B}_i(t) &= \frac{-B_i(t) + c_B \sum_j \delta(t - s_i^{(j)})}{\tau_B},\end{aligned}\tag{3.76}$$

where $s_i^{(j)}$ represents the j 'th spike time of the input spike train arriving at the i 'th synapse. In this equation, the variables A and B are used to generate a double exponential profile (see for instance [Rotter and Diesmann, 1999]). In this model, it holds that:

$$\begin{aligned}c_i(t) &= -\bar{g}(A_i(t) - B_i(t))E_r \\ d_i(t) &= \bar{g}(A_i(t) - B_i(t)).\end{aligned}\tag{3.77}$$

Excellent agreement with the NEURON simulator is achieved (Fig 3.2F). The amount of kernels required is far lower than in the normal GF formalism (Fig 3.2D) and optimal performance (for which we found $n_k \approx 7$) was achieved for $K = 3$ (Fig 3.2E).

3.3.1 Computational cost

It is immediately clear that the computational cost of the SGF formalism is far lower than the computational cost of the GF formalism [Wybo et al., 2013] from the number of required kernels alone (Fig 3.2D). However, the comparison of computational cost with the canonical finite difference approaches requires a more careful discussion. For n input locations and

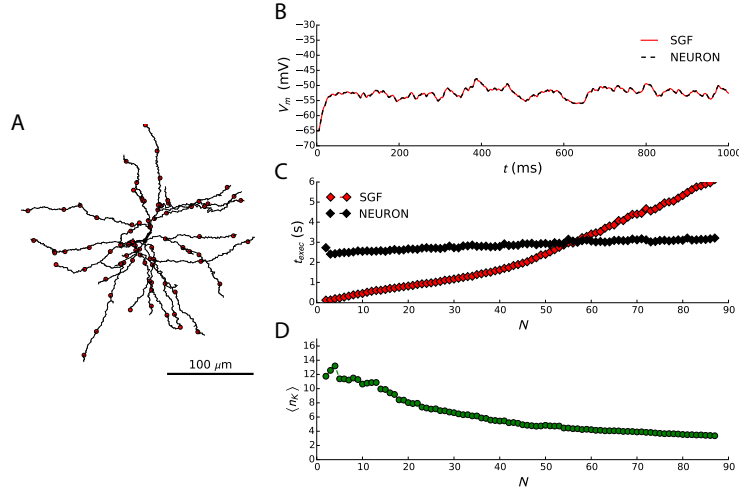


Figure 3.3 – **The execution time of the SGF formalism compared to a NEURON simulation.** **A:** The setup, showing the morphology and the maximal set of input locations $N = 74$ input location. Note that this set was chosen so that $|\mathcal{N}| = 2$ for all \mathcal{N} . **B:** The somatic voltage trace for the first second of a 10 s simulation, with a conductance based excitatory synapse at each input location indicated in A. **C:** Execution times as a function of the number of input locations. **D:** The average number of operations per kernel as a function of the number of input locations.

n_t time steps, an explicit solver would typically require $O(n_k n_t n)$ steps, whereas a canonical finite difference approach would require $O(d_t n_t n_x)$ steps, with n_x the number of spatial locations at which the finite differences have to be evaluated and d_t the maximal degree of temporal differentiation in the operators $\hat{L}_i^{(t)}(x)$. Thus, we expect the SGF formalism to improve performance when $cn_k n < d_t n_x$, where c is an a-priori unknown constant that may depend on the specific implementation. When the system is stiff, an implicit solver is typically needed to guarantee stability. Such solvers require a matrix of size n (resp. n_x in the case of finite differences) to be inverted each time step, normally an operation of complexity $O(n_x^3)$. However, for second order finite difference solvers, the tree graph introduces a special structure in this matrix [Hines, 1984], so that it can be inverted in $O(n_x)$ steps. In the SGF formalism, this structure can be maintained if $|\mathcal{N}| = 2$ for all \mathcal{N} and in this case the same performance criterion $cn_k n < d_t n_x$ applies as for explicit solvers. Note that for higher order finite difference approaches the matrix inversion still requires $O(n_x^3)$.

To test whether this constant c is not excessively large, so that it may impede the usefulness of the SGF formalism, we compared the execution time of the SGF formalism with the execution time of the NEURON simulator [Carnevale and Hines, 2006]. In order to obtain a ‘fair’ comparison, attention must be given to the exact type of implementation. For instance, comparing an explicit simulation paradigm with NEURON makes little sense, as NEURON implements a semi-implicit paradigm and would be severely at the disadvantage. On the other hand, NEURON makes use of the Hines algorithm to invert the systems’ matrix [Hines, 1984], and dis-

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

tributing input location completely at random in the SGF formalism (and using a semi-implicit paradigm as well) would exclude the use of this algorithm, which would severely disadvantage the SGF formalism. We therefor opted to implement a simulator in C++, that implements the same semi-implicit method described in equations (3.49) and (3.47), but where the inversion uses the Hines algorithm. This restricts the input locations to configurations where $|\mathcal{N}| = 2$ for all \mathcal{N} . We distributed 2 to 74 input locations in such a configuration (Fig 3.3A shows this configuration for $n = 74$) on the same stellate cell morphology used in Fig 3.2 (panels C-E). No active channels were added to the model. In NEURON, the cell model contained 299 compartments for a total dendritic length of approximately $4000 \mu\text{m}$ ($\Delta x \approx 13.5 \mu\text{m}$). We added one excitatory, conductance based synapse at each input location and gave it a Poisson spike train of a certain rate, so that the total presynaptic rate at all synapses together was 1000 Hz. In Fig 3.3B we show the first second of a 10 s simulation where $n = 74$. It can be seen that both traces agree impeccably. We then compared the execution times of both models for an increasing number of input locations (Fig 3.3C) for a simulation time of 10 s at a time step of 0.1 ms. It can be seen that the SGF implementation outperforms the NEURON model until $n \approx 55$ (on a machine equipped with an Intel Core i7-3770 CPU @ 3.4GHz and 16 GB of RAM, running ubuntu 14.04 LTS). In Fig 3.3D, we plotted $\langle n_k \rangle$, the average number of operations per kernel and per time step. It can be seen that this number correlates with the execution time. When there are few input locations, kernels are in general longer and hence this number is higher. Then, when more kernels are added, this number drops quite rapidly. This explains the small slope of the execution times in the SGF formalism until $n \approx 35$, associated with the rapid decrease of $\langle n_k \rangle$. After that, $\langle n_k \rangle$ decreases slower, and hence the slope of the execution times becomes steeper.

3.4 Summarizing remarks

We have proven that on tree graphs, the GF formalism for linear, non-homogeneous, time-invariant PDE's with inputs at a discrete set of n well-chosen locations requires only $O(n)$ rather than n^2 kernels, and termed this the SGF formalism. We discussed the meaning of 'well-chosen', namely that the sizes of the sets of nearest neighbors must be small (ideally $|\mathcal{N}| = 2$ for all \mathcal{N}). We have shown furthermore for equation (3.3) that in the limit of a small distances between the input locations, the SGF formalism can be reduced to the second order finite difference approximation. Thus, in some sense, the SGF formalism can be seen as a generalization of the second order finite difference approximation to arbitrary distance steps (as long as what lies between the input locations is approximately linear). We also employed the VF algorithm [Gustavsen and Semlyen, 1999], that fits the kernels with sums of exponentials, to design an efficient simulation algorithm.

We then validated our algorithm on two neuroscientific problems: the modeling of myelinated axons and of dendritic integration. We showed that there was excellent agreement between the SGF formalism and the de facto standard NEURON simulator. We found furthermore that, even for complex, branched morphologies, excellent accuracy was achieved for $n_k \lesssim 15$. We

also discussed when an increase in computational performance is expected from the SGF formalism, and furthered our findings by comparing an efficient C++ implementation of the SGF formalism with the same model in NEURON. Finally, we discuss in a broader sense when the SGF formalism is expected to provide advantage over other approaches.

3.5 Discussion

It can be seen that the question whether one should prefer the SGF formalism over second order finite difference approaches depends on the values of n_k and n , which are not necessarily independent. Fig 3.3 illustrates this: for the same dendritic arborization, and using the passive cable equation ($d_t = 1$), n_k ranges from ≈ 14 for $n = 2$ to ≈ 4 for $n = 74$. This leads to $n_k n \approx 28$ to 296 operations per time step. The dendritic arborization on the other hand has a total length of approximately $4000 \mu\text{m}$. According to the Lambda rule [Carnevale and Hines, 2006], the canonically accepted method of calculating the distance step between compartments, $\Delta x \approx 10 - 15 \mu\text{m}$. Consequently, a second order finite difference approximation would use approximately $n_x \approx 300$. Hence this estimate would indicate that performance can be gained when $n \lesssim 75$. That we found this number to be slightly lower ($n \lesssim 55$) in the simulations we conducted (Fig 3.3) may be due to implementation and optimization related factors. One important remark is that for efficient semi-implicit simulations, the input locations must be chosen so that $|\mathcal{N}| = 2$ for all \mathcal{N} to be able to use the Hines algorithm. The question may now be asked whether this is overly restrictive. We believe that this is not the case: placing an input location at the soma for instance separates all main dendrites. Bifurcations in the dendrites may induce further sets of nearest neighbors of sizes slightly larger than 2. Whenever this occurs however, an input location can be added at the bifurcation point, so that such a set is split as well.

Thus the choice of the SGF formalism over the finite difference approximation depends on the neural system at hand. In some systems, such as bipolar neurons used in auditory coincidence detection [Agmon-Snir et al., 1998, Wybo et al., 2013], inputs occur only at a small set of locations, and the cells' computation is performed by linear membrane properties. Other systems, such as myelinated axons, have non-linear membrane currents that are concentrated at a discrete set of locations – the nodes of Ranvier, with stretches of myelinated fiber in between that behave approximately linear. It is to be expected that in such systems, the SGF formalism may significantly improve performance over the second order finite difference approach. In other systems, such as cortical cells, inputs are distributed throughout the dendritic arborization in an almost continuous fashion. In these cells, the answer to the question whether the SGF formalism yields computational advantage is negative, when one aims at retaining all biophysical detail. Indeed, it is clear that when the number of input locations in the SGF formalism approaches the number of compartments NEURON requires, there is no computational gain in using the former. Nevertheless, computational neuroscientists have been seeking ways of reducing the cost of simulating these cells to be able to use them in large scale network simulations. Most often, they aim to achieve this by drastically reducing

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

the number of compartments [Traub et al., 2005]. This approach however also changes the spatio-temporal response properties of the nerve cells. With the SGF formalism, inputs that would otherwise be grouped in a small number of compartments may now be grouped at a small number of input locations, while the response properties induced by the neuronal morphology would remain unchanged. Finally, in dendritic arborizations, a number of resonance phenomena have been observed that can be modeled with linearized (quasi-active) ion channels [Koch, 1984, Laudanski et al., 2014]. Since the transitivity property (see appendix §3.6.2)) holds for equations of this type as well (of the general form (3.8)), such systems can be modeled implicitly in the SGF formalism. Nevertheless, the validity of this linearization depends on the size of the fluctuations around the operating point and the ion channel under study and has to be checked.

In the SGF formalism there is a significant initialization phase before a neuron model can be simulated (see Implementation). The computational cost of this phase is higher than the cost of the initialization phase for finite difference approaches, and depends on the complexity of the tree graph and the number of input locations. We thus expect that our SGF formalism will be advantageous in use cases where frequent re-initialization of the model is not required. This is typically the case for network simulations, where a limited number of prototypical nerve cells may be initialized and used throughout the network.

Another important matter, next to computational cost, is the accuracy of the SGF formalism. While the sparsification is exact, the transform back to the time-domain along with the specific integration algorithm might introduce errors. The use of the approximate VF algorithm however impedes a systematic analysis of these errors. Nevertheless, we found that typically this error was very low (Fig 3.4B,C). Furthermore, after application of the integration paradigm described in this work, where the convolutions with exponentials are integrated analytically assuming that the voltage varies linearly in between grid points, we found that all our numerical experiments agreed very well with equivalent NEURON simulations.

Acknowledgments

We thank Luc Guyot for many insightful comments on the manuscript. The research leading to these results has received funding from the European Union - Seventh Framework Programme [FP7/2007-2013] under grant agreements no. 269921 (BrainScaleS) and no. 604102 (The HUMAN BRAIN PROJECT) as well as funding from the EPFL Blue Brain Project Fund and the ETH Board to the Blue Brain Project.

Author contributions

WW and BTN designed the research. WW and DB did the math. WW did the simulations and analyzed the results. WW, BTN and MOG wrote the paper.

3.6 Appendix

3.6.1 Proof of Lemma 1

Proof. **The PDE defined on a line**

Consider a line of length L ($0 \leq x \leq L$), which trivially has two leafs ($\Lambda = \{\lambda_1, \lambda_2\}$). Fourier transforming a PDE such as (3.8) leads to a boundary value problem of the form:

$$\tilde{L}_0(x, \omega) \frac{\partial^2}{\partial x^2} \tilde{V}(x, \omega) + \tilde{L}_1(x, \omega) \frac{\partial}{\partial x} \tilde{V}(x, \omega) + \tilde{L}_2(x, \omega) \tilde{V}(x, \omega) = \tilde{I}(x, \omega), \quad 0 < x < L \quad (3.78)$$

$$\begin{aligned} \hat{B}^{\lambda_1} \tilde{V}(0, \omega) &:= \tilde{L}_1^{\lambda_1}(\omega) \frac{\partial}{\partial x} \tilde{V}(L, \omega) + \tilde{L}_2^{\lambda_1}(\omega) \tilde{V}(0, \omega) = \tilde{I}^{\lambda_1}(\omega) \\ \hat{B}^{\lambda_2} \tilde{V}(0, \omega) &:= \tilde{L}_1^{\lambda_2}(\omega) \frac{\partial}{\partial x} \tilde{V}(L, \omega) + \tilde{L}_2^{\lambda_2}(\omega) \tilde{V}(L, \omega) = \tilde{I}^{\lambda_2}(\omega), \end{aligned} \quad (3.79)$$

The Green's functions $g(x, x_0, \omega)$ is obtained from solving this problem for $\tilde{I}(x, \omega) = \delta(x - x_0)$ and $\tilde{I}^{\lambda_1}(\omega) = \tilde{I}^{\lambda_2}(\omega) = 0$, and is given in [Stakgold, 1967, page 66] :

$$g(x, x_0) = \begin{cases} \frac{u_1(x) u_2(x_0)}{a_0(x_0) W(u_1, u_2; x_0)}, & 0 \leq x \leq x_0 \\ \frac{u_1(x_0) u_2(x)}{a_0(x_0) W(u_1, u_2; x_0)}, & x_0 \leq x \leq L, \end{cases} \quad (3.80)$$

where $u_1(x)$ is a non-trivial solution of the homogeneous equation satisfying $\hat{B}^{\lambda_1} u_1 = 0$ and $u_2(x)$ a non-trivial solution satisfying $\hat{B}^{\lambda_2} u_2 = 0$, where $W(u_1, u_2; x)$ denotes the Wronskian of both solutions evaluated at x and where we have omitted the dependence on ω for clarity. From this equation, it can be checked that the **transitivity property** holds:

$$g(x_1, x_3) = \frac{g(x_1, x_2) g(x_2, x_3)}{g(x_2, x_2)} \quad (3.81)$$

when $x_1 \leq x_2 \leq x_3$ or $x_3 \leq x_2 \leq x_1$.

The generalization to a tree graph

Consider now the generalization of problem (3.78) to a tree graph. On each edge $e \in E$, an operator of the form:

$$\hat{L}^e(x) := \tilde{L}_0^e(x) \frac{\partial^2}{\partial x^2} + \tilde{L}_1^e(x) \frac{\partial}{\partial x} + \tilde{L}_2^e(x) \quad (3.82)$$

constrains the field V^e :

$$\hat{L}^e(x) \tilde{V}^e(x) = \tilde{I}^e(x), \quad (3.83)$$

where it is understood that x signifies the space coordinate on the edge under consideration.

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

On each leaf $\lambda \in \Lambda$ a boundary condition of the form (3.79) holds:

$$\hat{B}^\lambda \tilde{V}^\lambda = \tilde{I}^\lambda \quad (3.84)$$

and at each node that is not a leaf $\phi \in \Phi$:

$$\tilde{V}^\epsilon = \tilde{V}^{\epsilon'}, \quad \forall \epsilon, \epsilon' \in E(\phi) \quad (3.85)$$

$$\sum_{\epsilon \in E(\phi)} \tilde{L}^\epsilon \frac{\partial}{\partial x} \tilde{V}^\epsilon = \tilde{I}^\phi, \quad (3.86)$$

Equation (3.85) assures continuity of the field and a condition of the form (3.86) is imposed in many physical systems to assure the conservation of flow.

We wish to determine Green's function $g(x, x_0)$ of this problem. When x_0 is located on edge ϵ_0 , we need to solve this problem for $I^\epsilon(x) = \delta_{\epsilon\epsilon_0} \delta(x - x_0)$, $I^\lambda = I^\phi = 0$. We will first show that at each end of edge ϵ_0 boundary conditions of the form (3.79) hold, by using (3.84), (3.85) and (3.86).

To do so, we only need the following recursion rule: Consider a node ϕ , and suppose that all but one of the edges in $E(\phi)$ satisfy a boundary condition $\hat{B}^\epsilon \tilde{V}^\epsilon = 0$ of the type (3.79) at the opposite end of node ϕ . Within each edge, we chose the spatial coordinate x^ϵ to be L^ϵ (i.e. the edge's length) at that far end and 0 at the node. Let us call the edge that does not satisfy the boundary condition ϵ' . Thus we have:

$$\hat{B}^\epsilon \tilde{V}^\epsilon = \tilde{L}_1^\epsilon \frac{\partial}{\partial x} \tilde{V}^\epsilon(L^\epsilon) + \tilde{L}_2^\epsilon \tilde{V}^\epsilon(L^\epsilon) = 0, \quad \forall \epsilon \in E(\phi) \setminus \epsilon', \quad (3.87)$$

and from (3.85) and (3.86) it follows that:

$$\tilde{V}^{\epsilon'}(0) = \tilde{V}^\epsilon(0), \quad \forall \epsilon \in E(\phi) \quad (3.88)$$

$$\sum_{\epsilon \in E(\phi) \setminus \epsilon'} \tilde{L}^\epsilon \frac{\partial}{\partial x} \tilde{V}^\epsilon(0) + \tilde{L}^{\epsilon'} \frac{\partial}{\partial x} \tilde{V}^{\epsilon'}(0) = 0, \quad (3.89)$$

We will show that from conditions (3.88) and (3.89) a boundary condition of the form (3.79) can be derived for $\tilde{V}^{\epsilon'}(0)$ (i.e. the field on edge ϵ' at the location of node ϕ), as long as the differential equations on edges $\epsilon \in E(\phi) \setminus \epsilon'$ are homogeneous. Let $u^\epsilon(x)$ be a non-trivial solution of the homogeneous problem (3.83) on edge ϵ that satisfies condition (3.87). Every solution on that edge is necessarily of the form $\tilde{V}^\epsilon(x) = A^\epsilon u^\epsilon(x)$. As a consequence of (3.88), $A^\epsilon = \frac{\tilde{V}^{\epsilon'}(0)}{u^\epsilon(0)}$, which leads to a constraint on the derivative $\frac{\partial}{\partial x} \tilde{V}^\epsilon(0) = \frac{\tilde{V}^{\epsilon'}(0)}{u^\epsilon(0)} \frac{\partial}{\partial x} u^\epsilon(0)$. Thus, equation (3.89) becomes:

$$\left(\sum_{\epsilon \in E(\phi) \setminus \epsilon'} \tilde{L}^\epsilon \frac{\frac{\partial}{\partial x} u^\epsilon(0)}{u^\epsilon(0)} \right) \tilde{V}^{\epsilon'}(0) + \tilde{L}^{\epsilon'} \frac{\partial}{\partial x} \tilde{V}^{\epsilon'}(0) = 0, \quad (3.90)$$

precisely the boundary condition for $\tilde{V}^{\epsilon'}$ we were after.

Applying this operation recursively throughout the tree, starting from the leafs until edge ϵ_0 , assures that this edge has a boundary condition of the form (3.79) at both ends. Thus, on this edge, $g(x, x_0)$ is of the form (3.80).

To prove the transitivity property (3.81) for two arbitrary points x_1 and x_3 on the tree graph, and for a point x_2 that is on the shortest path between x_1 and x_3 , we distinguish four cases.

1. x_1, x_2, x_3 **are on the same edge**. Since the segment has boundary conditions of the type (3.79), the Green's function may be constructed as in (3.80), and thus (3.81) holds.
2. x_1 **and** x_2 **on the same edge**, x_3 **is on a different edge**. Let ϕ be the node adjacent to the edge ϵ_0 on which x_1 and x_2 are located and on the shortest path between x_2 and x_3 . Necessarily, the Green's function at that point satisfies

$$g(\phi, x_1) = \frac{g(\phi, x_2)g(x_2, x_1)}{g(x_2, x_2)}, \quad (3.91)$$

which then determines the solution on the adjacent edges $\epsilon \in E(\phi) \setminus \epsilon_0$ (where we choose the spatial coordinate x^ϵ to be 0 at ϕ and L^ϵ at the opposite end). On either of these edges, the solution is of the form $A^\epsilon u^\epsilon(x^\epsilon)$, with $u^\epsilon(x^\epsilon)$ a solution satisfying the derived condition of type (3.79) at the opposite end. Condition (3.85) then imposes $A^\epsilon = \frac{g(\phi, x_1)}{u^\epsilon(0)} = \frac{g(\phi, x_2)g(x_2, x_1)}{g(x_2, x_2)u^\epsilon(0)}$, and thus:

$$\begin{aligned} g(x^\epsilon, x_1) &= \frac{g(\phi, x_1)}{u^\epsilon(0)} u^\epsilon(x^\epsilon) = \frac{g(\phi, x_2)g(x_2, x_1)}{g(x_2, x_2)u^\epsilon(0)} u^\epsilon(x^\epsilon) \\ &= \frac{\frac{g(\phi, x_2)u^\epsilon(x^\epsilon)}{u^\epsilon(0)} g(x_2, x_1)}{g(x_2, x_2)} = \frac{g(x^\epsilon, x_2)g(x_2, x_1)}{g(x_2, x_2)}. \end{aligned} \quad (3.92)$$

We may apply this consideration recursively through the tree graph, until we arrive at the point x_3 , which proves relation (3.81) in this case.

3. x_2 **and** x_3 **on the same edge**, x_1 **is on a different edge**. Let ϵ denote the edge on which x_2 and x_3 are located, and let us denote the node adjacent to that edge at the side of x_1 by ϕ , and take the x -coordinate describing the position in that edge to be zero there ($x^\epsilon = 0$). Then $\tilde{V}^\epsilon(0) = g(\phi, x_1)$. On the other side of the edge, at $x^\epsilon = L^\epsilon$, the derived condition of the form (3.79) holds, and thus the field in that edge satisfies $\tilde{V}^\epsilon(x^\epsilon) = \frac{g(\phi, x_1)}{u^\epsilon(0)} u^\epsilon(x^\epsilon) (= g(x^\epsilon, x_1))$, where $u^\epsilon(x^\epsilon)$ is a solution satisfying this boundary condition. The Green's functions $g(x_2, x_2)$ and $g(x_3, x_2)$ are still of the form (3.80), as in this case the derived boundary conditions are valid on both ends of the edge. Let $v^\epsilon(x^\epsilon)$ be a solution that satisfies the homogeneous boundary condition at $x^\epsilon = 0$. Then, using

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

(3.80), the following holds:

$$\begin{aligned}
 g(x_3, x_2) \frac{1}{g(x_2, x_2)} g(x_2, x_1) &= \frac{v^\epsilon(x_2) u^\epsilon(x_3)}{a_0(x_2) W(u^\epsilon, v^\epsilon, x_2)} \frac{a_0(x_2) W(u^\epsilon, v^\epsilon, x_2)}{v^\epsilon(x_2) u^\epsilon(x_2)} \frac{g(\phi, x_1) u^\epsilon(x_2)}{u^\epsilon(0)} \\
 &= \frac{g(\phi, x_1) u^\epsilon(x_3)}{u^\epsilon(0)} \\
 &= g(x_3, x_1)
 \end{aligned} \tag{3.93}$$

4. x_1, x_2, x_3 **are on different edges.** This case is proved by combining the two previous cases.

□

3.6.2 Vector fitting

Here we briefly explain the VF algorithm [Gustavsen and Semlyen, 1999] as we implemented it. The version of this algorithm we needed approximates a complex function $f(s)$, for which $|f(s)| \rightarrow 0$ when $|s| \rightarrow \infty$ and $|f(s)| > 0$, as follows:

$$f(s) \approx \sum_{l=1}^L \frac{\gamma_l}{\alpha_l + s}, \tag{3.94}$$

where the parameter L is chosen. It does so in two steps: First the poles α_l are identified and then the residues γ_l are fitted.

Pole identification

First, a set of chosen starting poles $\tilde{\alpha}_l$ is specified, and an unknown auxiliary function $\sigma(s)$ with these poles is proposed, so that:

$$\sigma(s) = \sum_{l=1}^L \frac{\tilde{\gamma}_l}{\tilde{\alpha}_l + s} + 1 \tag{3.95}$$

$$[\sigma(s)f(s)]_{\text{fit}} = \sum_{l=1}^L \frac{\tilde{\gamma}_l'}{\tilde{\alpha}_l + s}. \tag{3.96}$$

Multiplying equation (3.95) with $f(s)$, and equating this with equation (3.96) gives:

$$\sum_{l=1}^L \frac{1}{\tilde{\alpha}_l + s} \tilde{\gamma}_l' - \sum_{l=1}^L \frac{f(s)}{\tilde{\alpha}_l + s} \tilde{\gamma}_l = f(s). \tag{3.97}$$

When f is evaluated at enough frequency points s (we found it sufficient to sample $f(s)$ on the imaginary axis $s \equiv i\omega_j, \omega_j \in \mathbb{R}, j = 1, \dots, N$ on an equidistant scale for small ω and on a logarithmic scale for large ω), this gives an over-determined system that can be solved

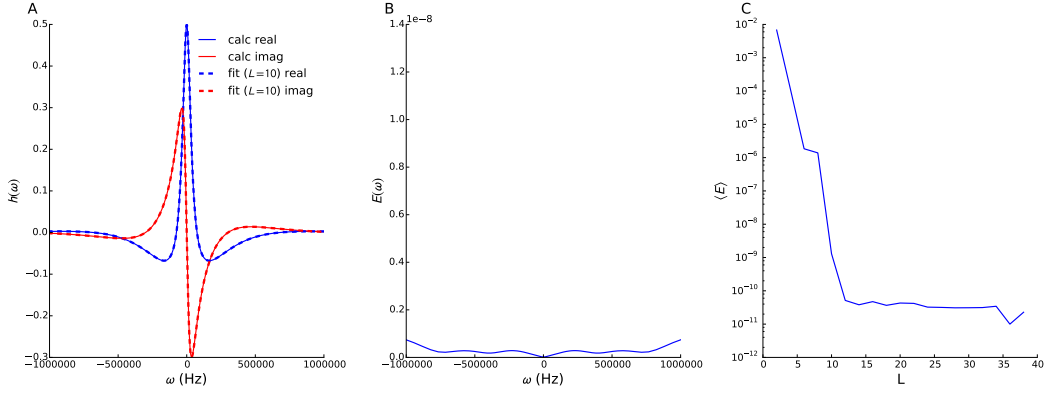


Figure 3.4 – **Illustration of the VF algorithm.** **A:** A typical kernel is accurately approximated by the VF algorithm (here $L = 10$). **B:** The error $E(\omega)$ of this fit. **C:** The average error of the fit as a function of L .

for $\tilde{\gamma}_l$ and $\tilde{\gamma}'_l$ by the least squares method. The poles of $f(s)$ are then given by the zeros of $\sigma(s)$ since, as the parametrization for $\sigma(s)$ satisfies equation (3.97), it holds that (see [Gustavsen and Semlyen, 1999, Hendrickx and Dhaene, 2006] for further details):

$$f(s) = \frac{[\sigma(s)f(s)]_{\text{fit}}}{\sigma(s)}. \quad (3.98)$$

These zeros can be found as the eigenvalues of the matrix:

$$H = \begin{pmatrix} \tilde{\alpha}_1 - \tilde{\gamma}_1 & -\tilde{\gamma}_2 & \dots & -\tilde{\gamma}_L \\ -\tilde{\gamma}_1 & \tilde{\alpha}_2 - \tilde{\gamma}_2 & \dots & -\tilde{\gamma}_L \\ \vdots & \vdots & \ddots & \vdots \\ -\tilde{\gamma}_1 & -\tilde{\gamma}_2 & \dots & \tilde{\alpha}_L - \tilde{\gamma}_L \end{pmatrix}. \quad (3.99)$$

Note that this procedure can be part of an iterative optimization, where the newly found poles can be used as the starting poles for the next iteration.

Residue fitting

Once the poles α_l are known, the residues can be determined by solving the over-determined system:

$$\sum_{l=1}^L \frac{1}{\alpha_l + s} \gamma_l = f(s) \quad (3.100)$$

for γ_l by the least squares method.

Chapter 3. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models

Accuracy

The VF algorithm does not provide the number of poles for the fit, nor does it provide an estimate for the accuracy of the fit with a given number of poles. We chose the number of poles as the smallest number for which the approximation gave a sufficient accuracy, defined as:

$$E(\omega_j) \equiv \frac{\left| f(i\omega_j) - \sum_{l=1}^L \frac{\gamma_l}{a_l + i\omega_j} \right|}{\max_j |f(i\omega_j)|} < \epsilon, \quad j = 1, \dots, N. \quad (3.101)$$

For our nerve models, we found that $\epsilon = 10^{-8}$ was sufficient. Usually, this accuracy was reached with $10 \lesssim L \lesssim 20$. In Fig 3.4A, we show a typical example of a kernel from the SGF formalism, together with its approximation with $L = 10$. In Fig 3.4B we show the error of this approximation as a function of the frequency, whereas Fig 3.4C we show how the average error, defined as:

$$\langle E \rangle = \frac{1}{N} \sum_{j=1}^N E(\omega_j), \quad (3.102)$$

changes with L .

4 Dynamic compartmentalization in neurons enables branch-specific learning

Willem A.M. Wybo, Benjamin Torben-Nielsen and Marc-Oliver Gewaltig, this manuscript is submitted.

Abstract

The dendritic trees of neurons play an important role in the information processing in the brain. Although it is widely accepted that dendrites require independent compartments to perform most of their computational functions, it is still not understood how they compartmentalize into functional subunits. Here we show how these subunits can be deduced from the structural and electrical properties of dendrites. We devised a mathematical formalism that links the dendritic arborization to an impedance-based tree-graph and show how the topology of this tree-graph reveals independent dendritic compartments. We find that balanced inputs or shunting inhibition can modify this topology and hence reconfigure the number and size of compartments in a context-dependent manner. We also find that dynamic re-compartmentalization can enable branch-specific learning of stimulus features.

4.1 Introduction

Brain function emerges from the collective behavior of billions of individual neurons that transform electrical inputs into action potential (AP) output. This transformation starts on the dendritic tree, where inputs are collected, and proceeds to the axon initial segment, where APs are generated that are then transmitted to downstream neurons through the axonal arborization. While axons appear to merely communicate the neuronal output downstream, dendrites collect and non-linearly transform input. This phenomenon is referred to as dendritic computation, and recent experimental work not only demonstrated that it occurs in vivo, but also that it is required for normal brain function [Grienberger et al., 2015, Takahashi et al., 2016, Smith et al., 2013].

Dendritic computations are plentiful [Rall et al., 1995, Segev, 2000, London and Häusser, 2005, Torben-Nielsen and Stiefel, 2010, Silver, 2010] and almost always require the existence of separate electrical compartments: regions on the dendritic tree that integrate inputs independently from others. Indeed, independent compartments enable branch-specific control of AP back-propagation [Müllner et al., 2015] as well as local N-methyl D-aspartate (NMDA), Ca^{2+} or Na^{+} spikes [Larkum et al., 2007, Wei et al., 2001]. They allow plasticity to operate on individual synapses with [Sjöström et al., 2008] or without [Golding et al., 2002] back-propagating APs and they allow individual neurons to function as multilayer neural networks [Poirazi et al., 2003b]. Moreover, they enable different input streams to be discriminated from each other [Johanning et al., 2009]. Thus, neurons require independent compartments to perform most of their theoretically predicted and experimentally demonstrated computational functions. Surprisingly, it is still not understood how dendrites compartmentalize into functional subunits and how these compartments can be deduced from the structural and electrical properties of dendrites.

4.2 Results

4.2.1 The neural evaluation tree

We investigate this compartmentalization using impedance-based tree-graphs. To obtain these tree-graphs, we compute the Green's function (GF) associated with a given morphology [Wybo et al., 2015] using the separation of variables method ([Major et al., 1993a], § 4.3.5). Then, by integrating over the temporal coordinate of the GF, we compute the impedance matrix, which expresses the electrical connectedness of regions on the dendritic tree (Fig 4.1A-B). On any neuron, all regions share at least some impedance, which will be associated with the root node in our tree-like network. Electrically closer regions will share more impedance, associated with an intermediate node. Finally, the leaf nodes represent impedance that is unique to non-overlapping regions. We visualize the impedance associated with a node as the length of the vertical line connecting it with its parent (Fig 4.1C). Importantly, we also developed a procedure that associates a temporal kernel with each node (whose surface is the nodal impedance, § 4.4.2), and an integration procedure that allows the computation of the full spatio-temporal voltage response (§ 4.4.3). The tree obtained in this way yields the voltage in all regions of interest (Fig 1D-E), and hence we term it the neural evaluation tree (NET). Thus, unlike previous work [Poirazi et al., 2003b, Behabadi and Mel, 2014], our networks are based on analytic calculations and time-resolved dynamics that capture both temporal and spatial interactions between synaptic inputs.

4.2.2 The impedance-based independence index

The NET allows systematic evaluation of whether or not input sites can function independently (§ 4.6). This is shown in Fig 4.1F-K, where we generated two NETs with different input configurations: two synapses were either inserted on distinct branches (Fig 4.1F-H) or on

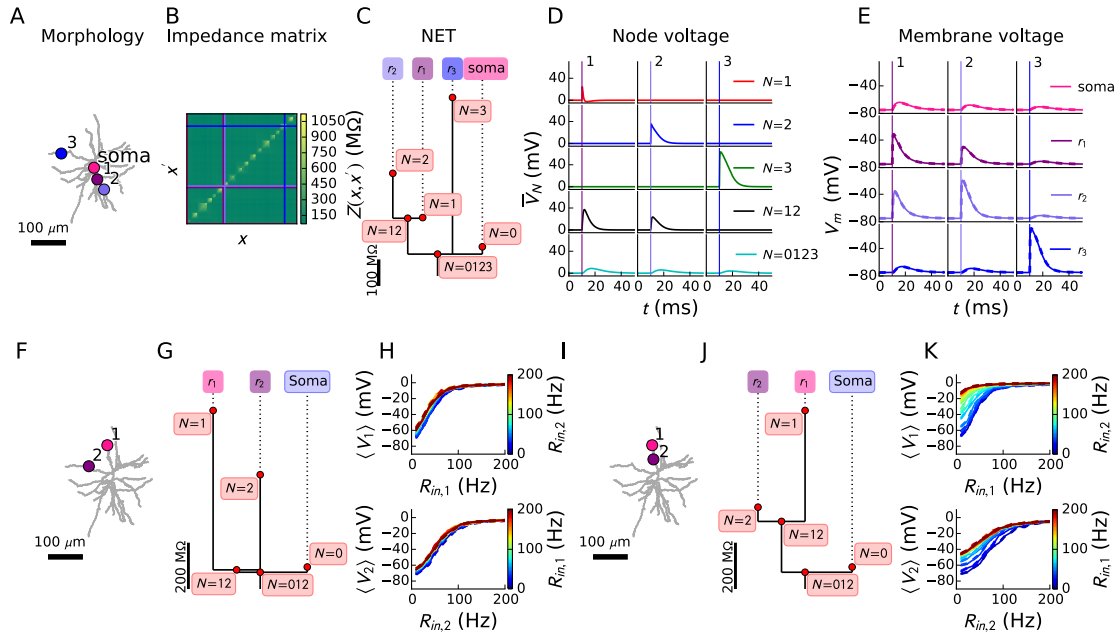


Figure 4.1 – Constructing a neural evaluation tree (NET) allows for the systematic evaluation of crosstalk between input sites. Panels A-E illustrate how the NET computes the voltages and panels F-K show how the NET allows for evaluation of crosstalk between input sites. **A:** Dendrite with three inputs regions (numbers 1-3) and somatic readout. **B:** Impedance matrix calculated using the Green's function formalism. Color encodes the impedance between each pair of points. Horizontal and vertical lines indicate the regions from (A). **C:** A NET is extracted from the impedance matrix. Nodes contain voltages and edges are proportional to the impedance between the nodes. **D:** Voltages at each node in the NET are computed. Leaf nodes ($N = 0, 1, 2, 3$) contain voltage components unique to the associated regions while lower nodes ($N = 12$ and $N = 0123$) contain voltage components that are shared between regions. Voltages contained in each node are shown after an input is presented in location 1 (left), 2 (middle) and 3 (right). Onset of the input is indicated with a vertical line in a color matching with the inputs location in (A). **E:** Final voltages calculated for the soma and input regions by summing the node voltages on the path from root to associated leaf. Dashed lines are computed using the NET and solid lines are the voltages from simulating an equivalent NEURON model. **F, I:** Two synapses are placed on the morphology, once on different branches (F) and once on the same branch (I). **G, J:** The respective NETs associated with the synapse configurations in F and I. **H, K:** On top, the two-dimensional response surface at synapse 1 as a function of $R_{in,1}$, the input rate at synapse 1. At the bottom, the response surface at synapse 2 as a function of $R_{in,2}$, the input rate at synapse 2. The color-coding indicates the input rate at the other synapse (2 on top and 1 at the bottom). For mutually independent synapses the lines of different color would coincide as the response at synapse 1 would not be influenced by the rate at synapse 2 and vice versa. Dashed lines are again NET simulations and solid lines equivalent NEURON simulations.

the same branch (Fig 4.1I-K). Then, we stimulated the synapses in both NETs with Poisson spike trains of varying frequencies (see § 4.7 for the simulation parameters) and observed the voltage response at each synapse (Fig 4.1H, K). When inputs are on distinct branches, the average voltage in one location almost independent of the input rate at the other location (Fig 4.1H). Contrastingly, if the two inputs are on the same branch, the average voltage at either location is strongly modulated by the input frequency at the other location (Fig 4.1K). This analysis illustrates that voltages in nearby regions depend on each other, while voltages at distinct branches behave independently. The NETs formalize this notion: independent regions are separated by large impedances (Fig 4.1G, long edges to leafs), whereas otherwise much of the impedance is shared (Fig 4.1J, long edges between internal nodes). We quantify this observation by introducing the impedance-based independence index I_Z , a pairwise ratio between local and shared impedances (§ 4.6.2). Higher values of I_Z indicate less shared voltage and thus more independent voltage dynamics between regions as compared to lower values of I_Z .

We now quantify at which value of I_Z a pair of regions on the dendrite constitute independent subunits. Intuitively, dendritic regions are electrically independent when (i) the voltage dynamics between those regions have little correlation (“correlation criterion” [Rabinowitch and Segev, 2006]), and (ii) local non-linear dynamics (such as regenerative currents or NMDA spikes) can be generated independently of global activity (“non-linearity criterion” [Behabadi and Mel, 2014]). We test these criteria in a toy NET with two leafs and one root, representing, for instance, a single trunk that bifurcates into two child branches (Fig 4.2A). In this toy NET we can vary I_Z at will. To assess the correlation criterion, we constructed a control model (independent NET or iNET), where both leaf nodes are completely independent, by replacing the voltage in the root node \bar{V}_R with a long-term average (§ 4.6.1). We drive both models with Poisson spike-trains that result in supra-threshold NMDA-dynamics (Fig 4.3A). As expected, the nodal voltage correlation in the model with disconnected branches is zero (Fig 4.2B, dots) and the correlation in the other model decreases with increasing I_Z (Fig 4.2B, rhombi). Correlations become negligible when $I_Z \geq 10$, and the root mean square error (RMSE) between somatic NET and iNET voltages also vanishes beyond that point (Fig 4.2B, red trace). To assess the non-linearity criterion, we activate the first branch (§ 4.3.6) with a conductance above the NMDA spike threshold (for which we can compute the steady-state depolarization using Newton’s iteration, § 4.4.3), and the other branch with a conductance of increasing strength. We first record the node voltage \bar{V}_2 associated with the second branch (Fig 4.2C). At low $I_Z (\lesssim 6)$, voltage spreads from the first branch to the second. As a result, only a small extra depolarization can be detected in branch 2 upon activating synapse 2. However, we observe no discontinuity in this branch which would indicate an NMDA spike [Major et al., 2008]. At higher I_Z , such a discontinuity appears (inset Fig 4.2C) and for $I_Z \geq 10$ it reaches over 80% of the amplitude of an isolated NMDA spike (Fig 4.2C). The amplitude of the discontinuity is furthermore principally determined by I_Z , and not by the precise activation at synapse 1 (color coding in Fig 4.2C). When recording the root voltage \bar{V}_R , we expect NMDA events in independent branches to sum perfectly. In other words, when synapse 1 and 2 are activated

above the NMDA spike threshold the surplus $\Delta \bar{V}_R$ should be equal to the voltage \bar{V}_R when only synapse 1 is activated (inset Fig 4.2D) and hence their ratio should be one. This ratio indeed converges to one (Fig 4.2D) when I_Z increases and reaches over 80% of its maximum for $I_Z \geq 10$. As the local voltage is the sum of root and leaf voltages, combining \bar{V}_R and \bar{V}_2 allows quantification of the influence both branches exert on each other during learning (Fig 4.3D-E, where we used the plasticity model described in [Clopath et al., 2010]). When branch 1 is active, less input conductance is needed in branch 2 to depolarize it above the potentiation threshold. For $I_Z \geq 10$, this decrease is less than 20% of its maximal value.

Taken together, we conclude that for $I_Z \geq 10$ pairs of dendritic regions can be considered independent. Moreover, while both structural and physiological properties determine I_Z , it is only I_Z that determines independence (Fig 4.3F-H). This notion is further reinforced by the observation that the membrane voltage correlation between different synapses decreases as a function of independently of the morphology (Fig S2E, S2G).

4.2.3 Dendritic compartmentalization

Next, we asked how many such regions could maximally coexist along a dendritic tree, as well as what their location would be. Using the NET, we constructed an algorithm that divides the dendritic tree into independent regions given a minimal I_Z (see § 4.6.3). The resulting compartmentalization is shown in Fig 4.2E-G for a cortical stellate cell (Fig 4.2E), a hippocampal granule cell (Fig 4.2F) and a cortical L5 pyramidal neuron (Fig 4.2G). For each example, we show the impedance matrix (left), the associated NET with compartments for $I_Z \geq 10$ colored (center left, see also Fig 4.4A-C for the NET impedance matrix approximations) and the original dendrite structure with compartments (center right). We validated the NETs equipped with somatic AP channels (see § 4.5) by comparing their respective somatic voltage traces to traces computed with an equivalent NEURON [Carnevale and Hines, 2006] model (Fig 4.2E-G, right panels). We observe a good correspondence between the voltage traces, confirmed by a low error and high AP coincidence factor (Fig S2F).

Note that in the pyramidal cell, the number of compartments for $I_Z \geq 10$ was far less than the number of dendritic terminals (Fig 4.2G), and in the stellate and granule cells we did not identify any compartments at all (Fig 4.2E-F, see also Fig 4.7). We thus arrive at a conundrum: why would main dendrites bifurcate elaborately, if these branches cannot function independently? And if they cannot be independent, how can learning and memory be compartmentalized at the single branch level [Losonczy et al., 2008, Weber et al., 2016]?

4.2.4 Dynamic compartmentalization

As neurons perform different input-output transformations at different moments in time, such as during up/down states [Wilson and Kawaguchi, 1996], we hypothesized that the number of compartments in dendrites can be modified dynamically by spatio-temporal input patterns.

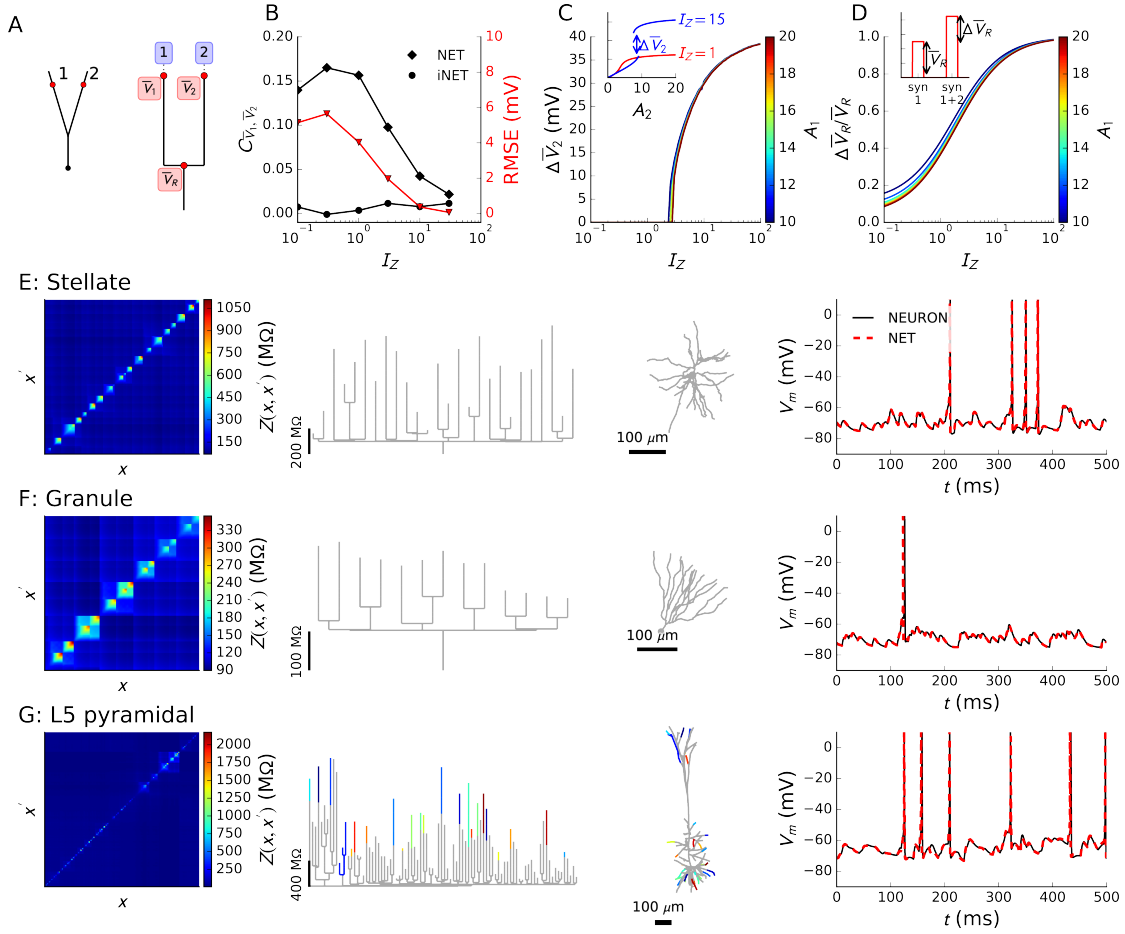


Figure 4.2 – **The impedance-based independence index I_Z leads to a systematic characterization of independent compartments.** **A:** Toy model dendrites and associated NET. **B:** The correlation between voltages in the leafs of NET and iNET (black), and their somatic RMSE (red), as a function of I_Z . **C:** With synapse 1 on, the jump in voltage $\Delta \bar{V}_2$ at NET leaf 2, indicating an independent NMDA-spike in branch two, is plotted as a function of I_Z . Color-coding indicates the strength of synaptic activation, which is expressed as the product between local input impedance and synaptic conductance (a dimensionless quantity, § 4.3.6). **D:** Similar configuration as in C, but now the ratio is shown between surplus voltage at the root when both synapses are on ($\Delta \bar{V}_R$) and the voltage when only synapse 1 is on (\bar{V}_R). **E:** Stellate cell. **F:** granule cell. **G:** Layer 5 pyramidal cell. For each cell, we indicate the exact impedance matrix (left), the NET with color coded independent regions at $I_Z = 10$ (center left, see materials and methods §4.3), the dendrite structure with color coded electrical compartments (center right), and a comparison between the time-resolved voltages at the soma calculated with the NET and with an equivalent compartmental model of the complete dendrite structure (right). The NETs (extended with linear layers, Materials and Methods §3) thus also handle supra-threshold responses.

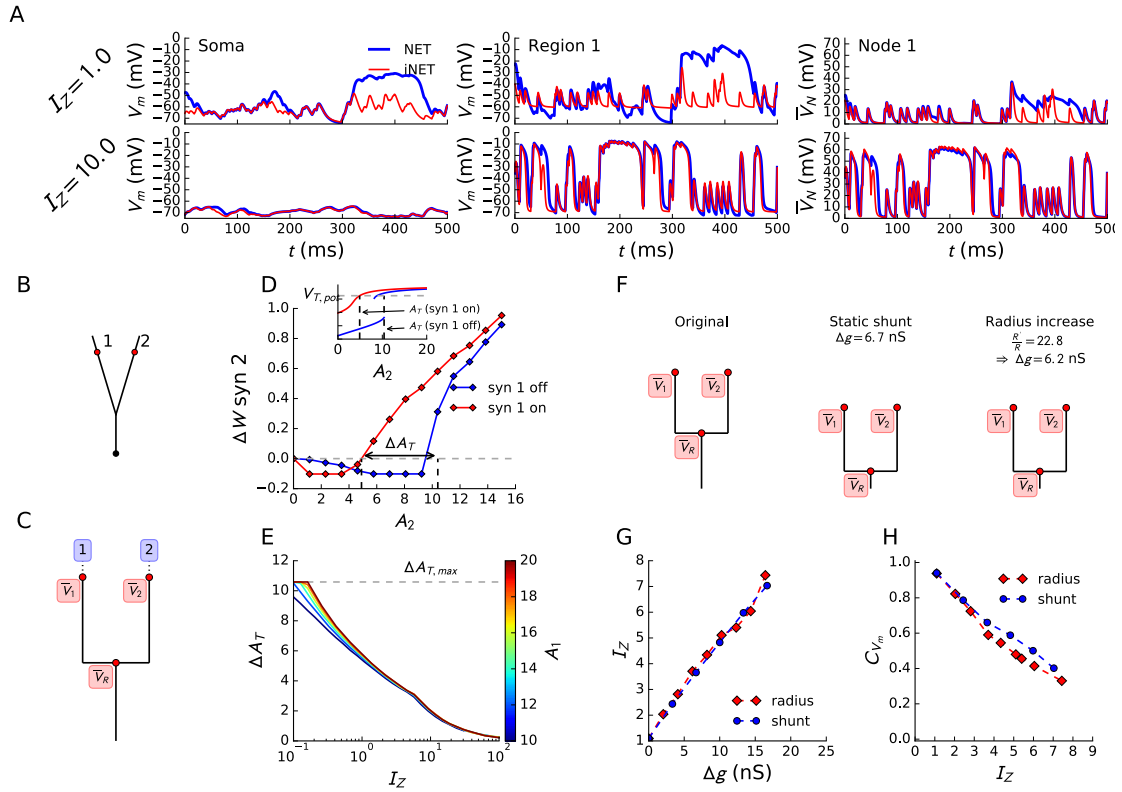


Figure 4.3 – Additional analysis for the two-branch toy model. **A:** Comparison between the NET and the iNET. Two of the traces associated with Fig 4.2B are shown, for $I_Z = 1$ (upper row) and $I_Z = 10$ (lower row). The left column shows the somatic voltage traces (in our simple model this is simply the potential of the root node), the middle column the total potential at one of input regions (r_1) and the right column the voltage at node 1. The normal NET trace is shown in blue and the iNET trace in red. For $I_Z = 1$, the iNET is a poor approximation of the real somatic potential, as most of the non-linear transformation happens in the root node. For $I_Z = 10$, most of the non-linear transformation happens in the leaf nodes and the iNET is a good approximation. **B:** The toy model morphology with a parent branch that bifurcates into two child branches. **C:** The NET associated with the morphology in B. **D:** Simulation of the weight change after 300 ms at synapse 2 when synapse 1 was off (blue) and on (red) ($I_Z = 1$). The computed shift in threshold activation required for potentiation at synapse 2 (inset) is indicated. **E:** The shift in threshold activation as a function of I_Z . Perfect independence would imply that there is no shift. The maximal shift is indicated in grey and indicates that if I_Z is very small, the voltage at synapse 2 will always be above the threshold for potentiation when synapse 1 is on. **F:** We modify I_Z by either adding a static shunt conductance to the parent branch or by increasing its radius. The original NET is shown on the left, the NET for an additional static shunt conductance in the middle and the NET for an increased radius on the right. Note that modifying I_Z by increasing the radius is not realistic in this case: to obtain a sizable increase in I_Z , the radius had to be multiplied by an unrealistically high factor. **G:** The increase in I_Z by the aforementioned manipulations is determined by the increase in conductance in the parent branch. **H:** The correlation for both ways of increasing I_Z . The inputs were tuned to use the full range of the NMDA non-linearity.

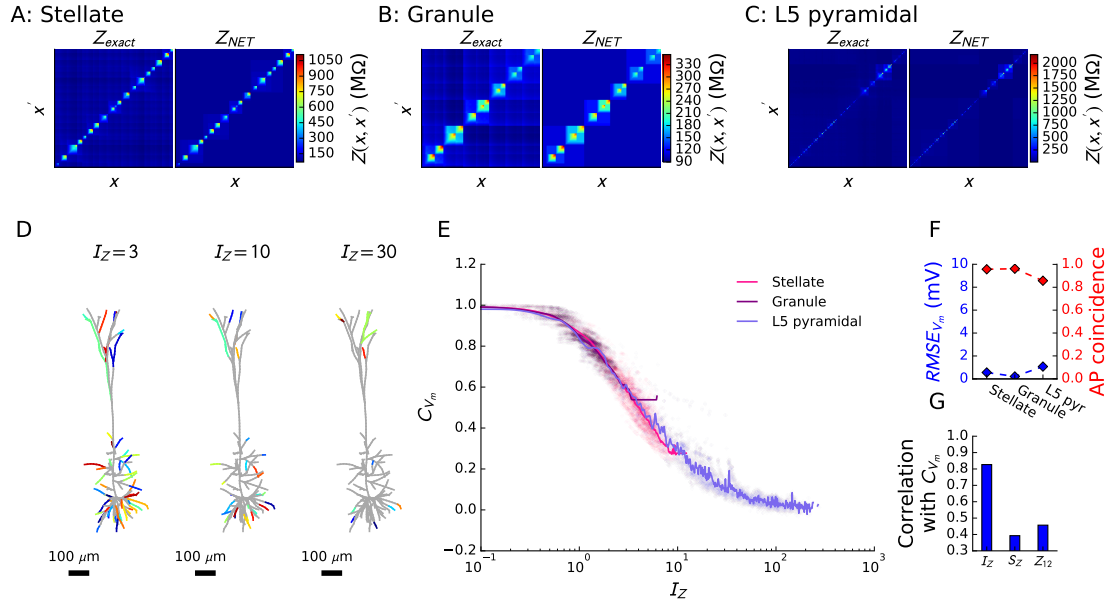


Figure 4.4 – Detailed analysis of the NET approximation for three cell morphologies. **A,B,C:** The true impedance matrix Z_{exact} versus the NET approximation Z_{NET} for resp. the stellate cell, the granule cell and the pyramidal cell. **D:** Analysis for the pyramidal cell of the change in compartmentalization when the minimum required separation I_Z between compartments is altered. **E:** The same simulations as in Fig 4.2E-G (right panels) were run for 100 s. For any pair of synapse locations the mutual voltage correlation C_{V_m} was computed and plotted as a function of their respective I_Z . A fit line was drawn using a kernel regression algorithm with radial basis functions. Independent of the neuronal type, C_{V_m} decreases as a function of I_Z . **F:** The root mean square error of the somatic sub-threshold potential (blue) as well as the AP coincidence factor for said simulation. **G:** The correlation of C_{V_m} with I_Z , a symmetry index S_Z (see Materials and methods §4.2) and the transfer impedance Z_{12} was checked. As can be seen, C_{V_m} is most correlated with I_Z .

To test this hypothesis, we first consider the high-conductance state [Destexhe et al., 2003]; a state occurring in vivo when many synapses are randomly activated, increasing the total membrane conductance. Fig 4.5A illustrates the contrast between the original NET of the neuron with input configuration depicted in Fig 4.1A (black) and the NET mimicking the high-conductance state (blue). A drastic difference is visible: the impedance at the root is strongly reduced. As a result, electrical separation becomes stronger between branches that only have the root in common, and compartments emerge in the stellate cell (Fig 4.2C). The high conductance NET is accurate in reproducing the average voltages: traces in Fig 4.2B (full colored lines) agree very well with the average post-synaptic potentials computed during an ongoing barrage of balanced excitation and inhibition (dashed colored lines).

Since both theory [Gidon and Segev, 2012] and connectivity data [Bloss et al., 2016] suggest the importance of the precise location of synaptic inhibition, we investigated the influence of few, precisely located inhibitory inputs on compartmentalization. In the apical tree of the L5 pyramidal neuron (Fig 4.2D) we noticed that the sibling branches in a particular apical fork did not constitute separate compartments (locations 3 and 4 in Fig 4.2E). Upon inserting inhibitory synapses near the branching point between the two terminal segments, they separate into independent compartments (Fig 4.2E, bottom panel). The required inhibition (with a time-averaged conductance of 15 nS) could easily be provided by one of the interneuron pathways targeting the apical tuft, such as the Martinotti or the bitufted to pyramidal cell pathways [Markram et al., 2015]. This change in compartmentalization can be quantified by the change in I_Z (Fig 4.2F). Note that the effect is location-specific: independence between locations 1 and 2 does not increase. In Fig 4.2G we verified these predictions by computing the correlation between the membrane voltages at locations where a marked decrease can be observed upon activation of the inhibitory synapse (3 and 4) and locations where this was not the case (1 and 2).

4.2.5 Branch-specific learning

Building on the ability of dendrites to dynamically rearrange independent subunits, we ask whether a transient recompartmentalization could improve branch-specific learning. Before learning, post-synaptic targeting is thought to be unspecific [Gerstner et al., 1996]. Hence, inputs coding different stimulus features can arrive at the same branch, but with different strengths. We ask whether sibling branches can learn to become selective only to the strongest initial feature (Fig 4.6A), using only NMDA spikes and no APs [Hardie and Spruston, 2009]. We tested this idea in two apical oblique sibling branches of the L5 pyramidal neuron (Fig 4.6A). If two branches receive different synaptic activation, the voltage difference between these branches will be larger when they are separated by a higher I_Z (Fig 4.6C), and hence the probability to robustly potentiate the preferred branch increases while the non-preferred branch is depressed. For the branches we choose (Fig 4.6A), I_Z was too low and this computation was not possible (Fig 4.6D, G) hence the evolution of synaptic weights in the preferred and non-preferred branches was positively correlated (Fig 4.6F). Activation of inhibitory synapses near

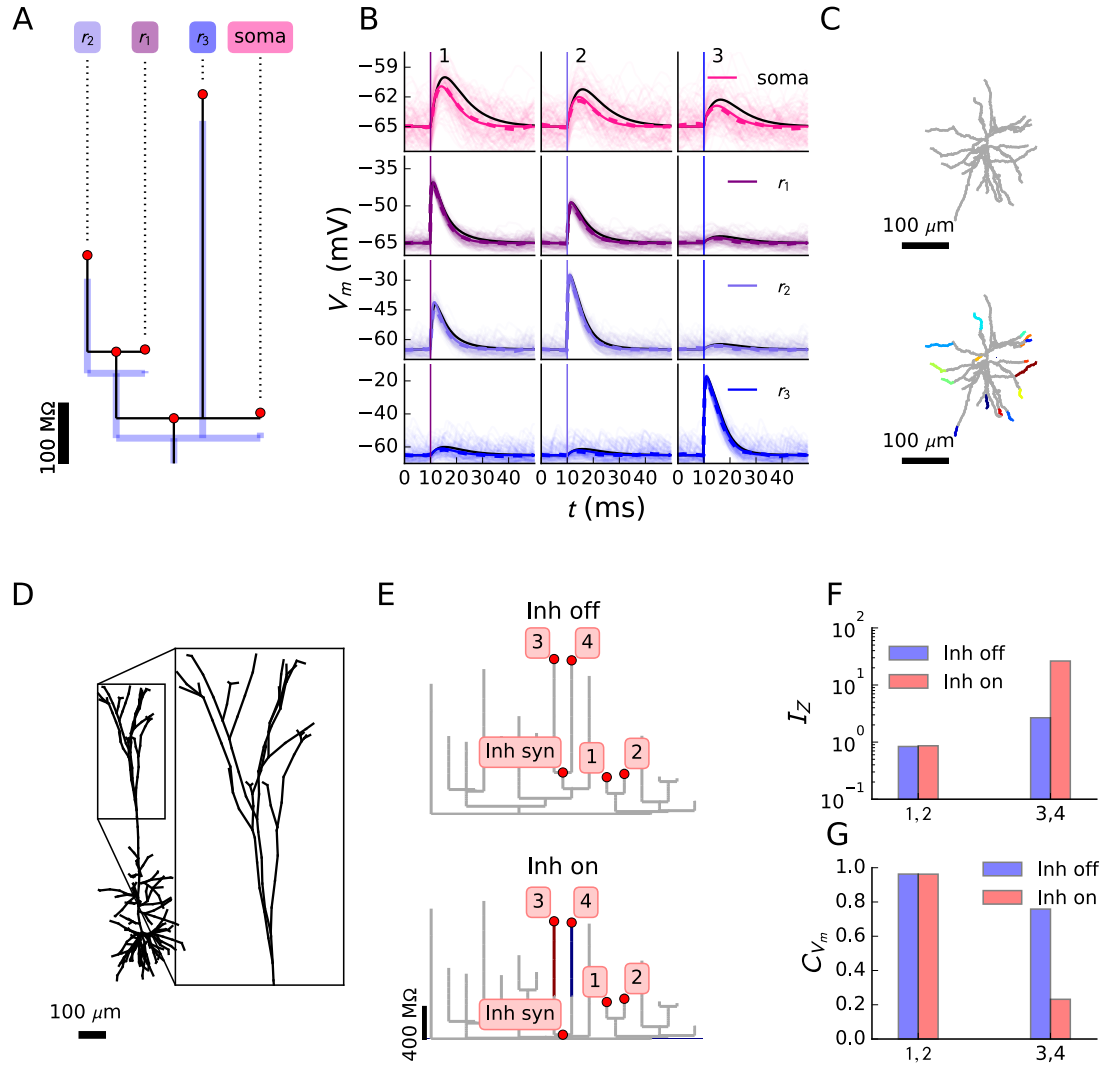


Figure 4.5 – Dynamic compartmentalization due to spatio-temporal input patterns. **A:** same NET as in Fig 1A,C (black). To mimic the high-conductance state, 200 excitatory and inhibitory synapses were distributed randomly on the neuron and activated with Poisson spike-trains. The NET was recomputed with the time-averaged synaptic conductances as static shunts (blue). **B:** At the three regions of interest, a strong excitatory synapse was inserted. The average post-synaptic potential was computed over 100 trials (dashed line) and coincides with the NET prediction (full line). The original responses are plotted in black for reference. **C:** Compartmentalization for $I_Z \geq 10$ in the rest state (top) vs. the high conductance state (bottom). **D:** The effect of inhibition on compartmentalization in the apical tuft of the L5 pyramidal cell was studied. **E:** NET associated with the apical tuft, without (top) and with (bottom) inhibition (with an time-averaged conductance of 15 nS). **F:** I_Z change when inhibition is turned on and **G:** associated change in membrane correlation when synapses in both branches were stimulated with random Poisson trains.

the bifurcation point increased I_Z (Fig 4.6B, black tree vs light blue tree) and hence enabled the branch-specific learning (Fig 4.6E, G), and the weight evolution became anti-correlated (Fig 4.6F).

4.2.6 Conclusion

Taken together, these three observations solve our conundrum: First, input regimes that increase membrane conductance, such as up states, result in the emergence of subunits in cells that had none during down states (Fig 4.5A-C). Second, location specific inhibition allows the precise tuning of compartmentalization (Fig 4.5D-G). Third, these factors enable branch-specific learning of stimulus features (Fig 4.6).

How many branches on the dendritic tree lie within in the useful range for re-compartmentalization? In pyramidal cells, compartments numbers double between $I_Z = 10$ and $I_Z = 3$ and in smaller cells these numbers increase five-fold (Fig 4.7). This suggests that global conductance input increases the number of compartments by a factor of two to five, depending on the cell type. Inspecting pairwise independence, we determine that in pyramidal cells 5% to 10% of terminal pairs are separated by I_Z values between 1 and 10 (mainly terminals on the same main branches). In smaller cells, up to 70% of pairs fall within these values. On average, these pairs are made independent by time-averaged shunting conductances of 15 nS (Fig 4.7).

In this paper we re-compartmentalized dendritic branches through shunting inputs with time-averaged conductances of 15 nS and 31 nS during the relevant intervals. While these values seem high compared to in-vivo data [Haider et al., 2013], in many situations the conductance required for re-compartmentalization is much lower. As a rule of the thumb, the impedance \bar{Z} of a node that receives a conductance g will be reduced by a factor $1 + \bar{Z}g$. If the input impedance of a parent branch is 500 M Ω , a conductance of 10 nS is required to increase I_Z between all it's child branches by a factor of 6, whereas a conductance of 2 nS would already increase I_Z by a factor of 2. Furthermore, the shunting conductance would only have to be provided locally and during the time-window of the excitatory input, as is commonly the case in balanced networks [Vreeswijk and Sompolinsky, 1996, Vogels et al., 2011, Dehghani et al., 2016]. For a root impedance of around 100 M Ω , which commonly occurs in smaller interneurons, a conductance of 50 nS thus reduces the associated impedance by a factor of 6, and as a consequence I_Z between branches increases by the same factor. Such a conductance increase is hypothesized to occur during the high conductance state [Rudolph and Destexhe, 2003].

Across the brain, neurons take on a wide variety of dendritic morphologies. We have shown here for the first time how these dendritic trees compartmentalize at rest and during dynamic input regimes. The behavioral relevance of up states (32) as well as the specificity of inhibitory targeting (28) suggest that dynamic compartmentalization is ubiquitous in normal brain function, with far-reaching consequences for memory formation [Kastellakis et al., 2016] and capacity [Wu and Mel, 2009].

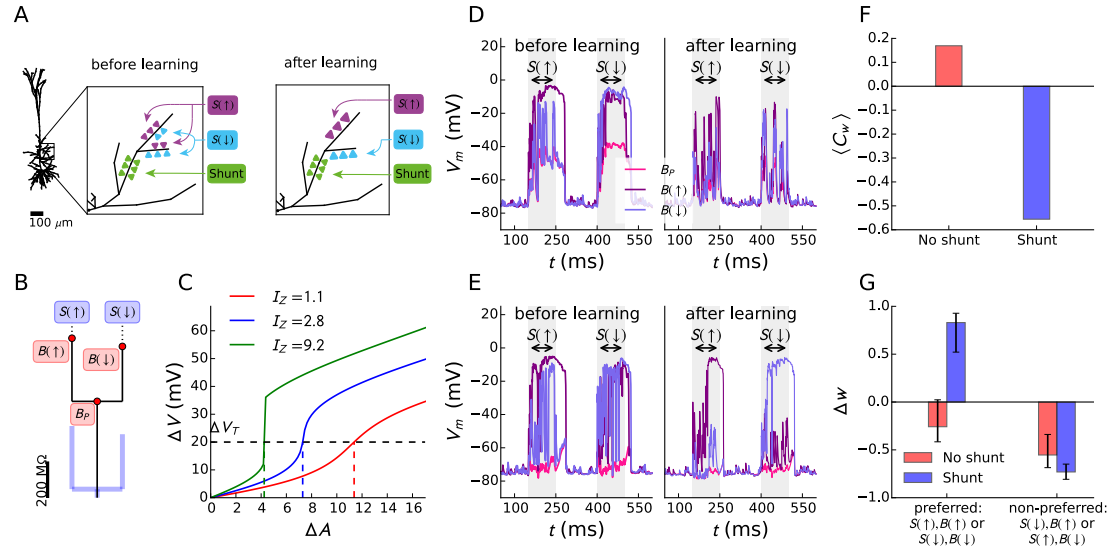


Figure 4.6 – **Tuning branch-specific learning with shunting conductances.** **A:** Sketch of the situation: before learning, two sibling branches are both targeted by synapses coding for either an up stimulus $S(\uparrow)$ (purple) or a down stimulus $S(\downarrow)$ (blue). One branch $B(\uparrow)$ receives more synapses of the former stimulus and the other branch $B(\downarrow)$ more of the latter stimulus. The branches should learn to be sensitive only to the stimulus that was initially the most prevalent (the 'preferred' stimulus). The shunting inhibition in the parent branch B_P had a time-averaged conductance of 31 nS when active. **B:** Schematic of the NET for the depicted situation. **C:** The difference in voltage across sister branches ΔV increases as a function of the difference in activation ΔA . For higher I_Z , this increase is steeper. Consequently, a given threshold ΔV (black dashed line) is reached for lower ΔA (colored vertical lines). **D, E:** The learning task without (D, effective $I_Z = 1.6$) or with (E, effective $I_Z = 10.5$) shunting inhibition. Each epoch, both stimuli are presented for 100 ms, with 150 ms intervals in between them, for a total of 20 epochs. The initial and final epochs are shown. **F:** Correlation between the average weights of the synapses in the preferred and non-preferred branches during stimulus presentation, averaged over all epochs and 20 repetitions of the learning task. **G:** Bar plot of the weight difference after the final epoch for all repetitions of the learning task. The bar lengths denote the medians of the weight difference distributions and the error bars the 25-75 percentiles. With shunt, the neuron successfully learns the task.

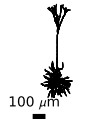
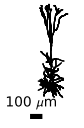
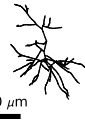


		M-type	L5_TTPC1 (2)	L5_TTPC2 (4)	L4_SS (4)	L23_NBC (5)	L23_BTC (3)
Morphometric	Example morphology						
	No. terminals		103.0±16.0	109.8±20.0	28.5±8.8	29.8±4.2	33.3±10.1
	No. main branches		8.5±0.5	8.2±0.8	5.5±0.5	7.6±1.6	4.7±1.2
	Total length (μm)		16045.7±1634.5	12391.0±2967.9	4020.3±815.4	3590.1±639.0	3692.5±813.0
Electrical	Z_{soma} (MΩ)		31.1±3.2	39.0±10.4	109.4±12.1	143.4±13.4	144.0±9.9
	Max Z_{in} (MΩ)		1871.3±570.3	1339.5±511.3	822.6±176.7	1229.1±174.8	2111.0±329.3
	Min Z_{trans} (MΩ)		4.5±0.4	7.7±4.8	65.2±17.3	99.6±22.5	84.2±24.6
NET	No. compartments ($I_Z=1$)		79.0±7.0	64.8±15.0	20.0±3.5	22.6±4.4	22.7±5.7
	No. compartments ($I_Z=3$)		59.0±7.0	44.5±10.3	12.2±4.9	12.6±4.8	11.7±2.6
	No. compartments ($I_Z=10$)		31.0±4.0	24.8±5.0	2.5±4.3	1.4±1.9	4.0±3.6
	No. compartments ($I_Z=30$)		6.5±1.5	9.5±6.7	0.0±0.0	0.0±0.0	0.0±0.0
	% terminal pairs $1 < I_Z < 10$		10.2±3.0	4.5±1.1	70.2±25.1	72.6±15.8	61.4±21.9
	Average I_Z		7.0±0.2	5.9±0.5	5.0±1.4	4.9±2.0	5.1±1.4
	Average g_S (nS) ($I_Z \rightarrow 10$)		15.2±3.2	21.5±4.0	14.8±6.0	13.1±8.2	12.1±5.0
	RMSE _Z (MΩ)		5.8±0.9	9.3±1.4	11.1±4.5	9.7±2.3	16.1±5.1

Figure 4.7 – **Analysis of five classes of cortical cells.** Five M-types from the BBP column were selected and analyzed in detail. Three groups of properties were computed: morphometric, electrical and NET-related properties. As morphometric properties we selected the number of terminal branches, the number of main branches emanating from the soma and the total dendritic length. As electrical properties we selected the somatic input impedance Z_{soma} , the maximal input impedance Z_{in} along the dendritic arborization and the minimal transfer impedance Z_{trans} between any pair of locations. Regarding the NET-related properties, we calculated the average number of compartments at different values of I_Z ($= 1, 3, 10, 30$). We also computed the percentage of terminal pairs that are separated by an I_Z between 1 and 10, the average I_Z of these pairs and the time-averaged shunting conductance required to increase the I_Z of these pairs to 10. To check whether the NET approximation yielded accurate results, we also computed the root mean square error RMSE_Z of the approximated impedance matrix with respect to the exact matrix. As can be seen, the error values we found are within similar ranges as those found for our three prototype cells (materials and methods, section 2.2). For each quantity, its average value and standard deviation is shown.

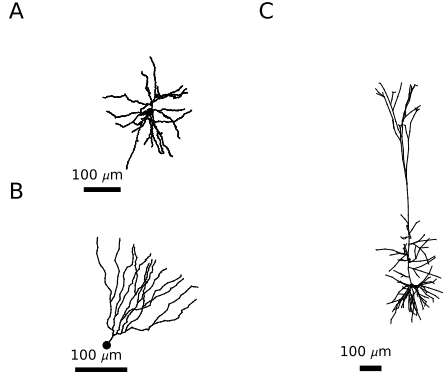


Figure 4.8 – **Cell morphologies.** **A:** a cortical layer 4 stellate cell (NeuroMorpho ID: NMO_00274), **B:** a hippocampal granule cell (NeuroMorpho ID: NMO_07645) and **C:** a cortical layer 5 pyramidal cell (ModelDB accession number: 139653). Note that branch thickness is not to scale.

4.3 Biophysical modeling

4.3.1 Morphologies

We centered our analysis around three morphologies: a cortical stellate cell (Fig 4.8A, [Wang et al., 2002]), a hippocampal granule cell (Fig 4.8B, [Carim-Todd et al., 2009]) and a cortical pyramidal cell (Fig 4.8C, [Hay et al., 2011]). These morphologies were retrieved from the NeuroMorpho.org repository [Ascoli, 2006], except the pyramidal cell, which was retrieved from the ModelDB repository [Hines et al., 2004]. Cell morphologies used in our wider cortical analysis were retrieved from the Blue Brain Project database [Markram et al., 2015].

4.3.2 Physiological parameters

Physiological parameters for the morphologies were set according to [Major et al., 2008]: the equilibrium potential was -75 mV, the membrane conductance $100 \mu\text{S}/\text{cm}^2$, the capacitance $0.8 \mu\text{F}/\text{cm}^2$ and the intracellular resistance $100 \Omega \cdot \text{cm}$.

To generate somatic action potentials (AP), we used the fast inactivating Na^+ current (I_{Nat}) and the fast, non-inactivating K^+ current ($I_{\text{Kv3.1}}$) employed by [Hay et al., 2011]. Channel densities were taken from one of their models on modelDB: $\bar{g}_{\text{Nat}} = 1.71 \text{ S}/\text{cm}^2$ and $\bar{g}_{\text{Kv3.1}} = 0.766 \text{ S}/\text{cm}^2$. The leak current was then fitted to yield a membrane time scale of 10 ms and an equilibrium potential of -75 mV.

Alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) and gamma-aminobutyric acid (GABA) synaptic input currents were modeled as the product of a conductance profile, given by a double exponential shape [Rotter and Diesmann, 1999], with a driving force [Jack et al., 1975]:

$$I_{\text{syn}}(t) = g(t) \cdot (E_r - V(t)). \quad (4.1)$$

For all synapses, time constants were chosen in accordance with canonically accepted values [Roth and van Rossum, 2009]. For AMPA synapses, we used rise resp. decay times $\tau_r = 0.2$ ms,

$\tau_d = 3$ ms for the conductance window and a reversal potential $E_r = 0$ mV, while for GABA synapses we used $\tau_r = 0.2$ ms, $\tau_d = 10$ ms and $E_r = -80$ mV. For N-methyl-D-aspartate (NMDA) channels [Jahr and Stevens, 1990], the synaptic current had the form

$$I_{\text{syn}}(t) = g(t) \cdot (E_r - V(t)) \sigma(V(t)) \quad (4.2)$$

and the rise resp. decay time were $\tau_r = 0.2$ ms, $\tau_d = 43$ ms, and $E_r = 0$ mV, while $\sigma(V)$ was the sigmoidal function employed by [Behabadi and Mel, 2014] to model the channels' magnesium block:

$$\sigma(V) = \frac{1}{1 + 0.3 e^{-0.1 V}}. \quad (4.3)$$

In the remainder of this work, we will refer to the voltage-dependent factors in the synaptic input current as the 'synaptic voltage dependence' (SVD), denoted by $f(V)$. Hence, for AMPA or GABA synapses

$$f(V) = E_r - V \quad (4.4)$$

and for NMDA synapses

$$f(V) = (E_r - V) \sigma(V). \quad (4.5)$$

Note that when we refer to the conductance of a simple synapse, we mean the maximum value g_{max} of its conductance window. For a synapse that has AMPA and NMDA components (to which we will simply refer as an NMDA synapse), the conductance is the maximal value of the AMPA conductance window, and the conductance of the NMDA component is determined by multiplying the AMPA conductance value with an NMDA ratio R_{NMDA} , that was set to be either 2 or 3 [Behabadi and Mel, 2014]. Note that none of our results depend on the precise NMDA/AMPA ratio.

4.3.3 Plasticity

In our simulations with plasticity, we use a voltage dependent spike timing dependent plasticity rule [Clopath et al., 2010, Bono and Clopath, 2016] where the evolution of the weight $w(t)$ of a given synapse depends both on the post-synaptic voltage and the presynaptic AP inputs. Next to modeling the dependence of the weight change on the pre and post spike timings [Clopath et al., 2010], this rule also models dendritic plasticity in the absence of back-propagating APs through its voltage dependence [Bono and Clopath, 2016]. In these simulations, we conceptualize the weight as a factor that rescales the conductance profile, leading to the following form for the synaptic current:

$$I_{\text{syn}}(t) = g(t) w(t) f(V(t)). \quad (4.6)$$

Chapter 4. Dynamic compartmentalization in neurons enables branch-specific learning

In all our simulations the initial weight $w(t = 0)$ was 1 and, during the simulation, the weight could fluctuate in the interval $[0, 2]$ according to the rule:

$$\frac{dw(t)}{dt} = -A_{LTD} X(t) (V_-^{LP}(t) - \theta_-)_+ + A_{LTP} X^{LP}(t) (V(t) - \theta_+)_+ (V_+^{LP} - \theta_-)_+, \quad (4.7)$$

where $(\cdot)_+$ is a rectifying function ($(x)_+ = x$ if $x > 0$ and otherwise $(x)_+ = 0$) and $V_-^{LP}(t)$ and $V_+^{LP}(t)$ are double low-pass filtered versions of the local voltage:

$$\begin{aligned} \tau_1 \frac{dV_1^{LP}}{dt}(t) &= -V_1^{LP}(t) + V(t) \\ \tau_+ \frac{dV_+^{LP}}{dt}(t) &= -V_+^{LP}(t) + V_1^{LP}(t) \\ \tau_- \frac{dV_-^{LP}}{dt}(t) &= -V_-^{LP}(t) + V_1^{LP}(t). \end{aligned} \quad (4.8)$$

Finally, $X^{LP}(t)$ is a low-pass filtered version of the pre-synaptic spike train $X(t) = \sum_{t_s} \delta(t - t_s)$:

$$\tau_X \frac{dX^{LP}}{dt}(t) = -X^{LP}(t) + X(t). \quad (4.9)$$

The constants in these equations are given by [Bono and Clopath, 2016]:

τ_1	5 ms	A_{LTP}	$3.5e-4$
τ_+	15 ms	A_{LTD}	$17.5e-4$
τ_-	45 ms	θ_+	-15 mV
τ_X	20 ms	θ_-	-74.3 mV.

4.3.4 Compartmental models

To construct and simulate compartmental models of the cells, we used the NEURON simulator [Carnevale and Hines, 2006]. Compartment sizes were set to be smaller than or equal to the size given by the lambda rule at 100 Hz [Carnevale and Hines, 2006].

4.3.5 Green's function and the separation of variables

To derive the neural evaluation tree (NET) approach that we will explain in the following section, we relied on the Green's function (GF) [Koch, 1998] $Z(x, x', t)$. The GF is a function of three variables: two locations x and x' along the dendritic arborization and a temporal variable t . We computed the GF in an exponential basis:

$$Z(x, x', t) = \sum_{k=0}^{\infty} \phi_k(x) \phi_k(x') e^{-\frac{t}{\tau_k}} \quad (4.10)$$

by using the separation of variables (SOV) method [Major et al., 1993a, Major and Evans, 1994]. Note that it is a property of the cable equation that the GF is symmetric in the spatial coordinates [Koch, 1998], so that $Z(x, x', t) = Z(x', x, t)$. Usually, a fixed set of discrete locations relevant for the problem at hand is chosen on the neuron. Hence, the GF only needs to be evaluated at these locations, and a discrete set of temporal functions, or ‘kernels’, is obtained. A member of this set will be denoted as $Z_{x,x'}(t)$, to highlight the difference between the now discrete indices x and x' and the continuous variable t .

To compute the output voltage $V_x(t)$ at location x for a given input current $I_{x'}(t)$ at location x' , one needs to compute the convolution of the GF evaluated at x and x' with this input current [Koch, 1998]:

$$V_x(t) = \int_0^\infty ds Z_{x,x'}(s) I_{x'}(t-s) \quad (4.11)$$

for which we will use the shorthand

$$V_x(t) = Z_{x,x'}(t) * I_{x'}(t). \quad (4.12)$$

Since $Z_{x,x'}(t)$ converts current into voltage, we will refer to it as an ‘impedance kernel.’ The total surface under the impedance kernel is the steady state impedance:

$$Z_{x,x'} = \int_0^\infty dt Z_{x,x'}(t). \quad (4.13)$$

In the rest of the text, it will be understood that $Z_{x,x'}$ without temporal coordinate refers to the steady state impedance – which we will simply call ‘the impedance’ for brevity – while $Z_{x,x'}(t)$ is the temporal impedance kernel. To unclutter the notations, we will not make this distinction for other variables; the temporal dependence will be omitted by default. Following this convention, equation (4.12) will be written as $V_x = Z_{x,x'}(t) * I_{x'}$, where it is implied that both $I_{x'}$ and V_x are time dependent quantities since $Z_{x,x'}(t)$ is the temporal impedance kernel. Conversely, writing $V_x = Z_{x,x'} I_{x'}$ means that $Z_{x,x'}$ is the steady state impedance value, and thus $I_{x'}$ and V_x will be steady state values too. Note that currents in this text will be expressed in nano ampere (nA) and voltages in milli volt (mV). Consequently, impedances will be in mega ohm (MΩ).

4.3.6 Synaptic activation

The eventual steady state voltage V_x obtained after activating a synaptic conductance at location x depends for a large part on the input impedance $Z_{x,x}$. Following (4.11), it can be obtained as a solution of the equation

$$V_x = Z_{x,x} g f(V_x). \quad (4.14)$$

This solution D is thus a function of the product of impedance and conductance for a given combination of synapse types:

$$V_x = D(Z_{x,x} g). \quad (4.15)$$

We refer to this product as the synaptic activation $A_x = Z_{x,x} g$ and note that it is a dimensionless quantity. Consequently, it is a convenient quantity that does not depend on local morphological constraints to determine whether an input will be strong enough to reach a certain voltage threshold, for instance to elicit an NMDA spike or to potentiate a synapse.

4.4 Neural evaluation tree

4.4.1 Introduction: two input regions

To explain the idea behind our neural evaluation tree (NET) approach, we consider a neuron with two regions receiving dendritic input. Using the GF, and following the notations described in § 4.3, such a neuron can be modeled as [Wybo et al., 2013]:

$$\begin{aligned} V_1 &= Z_{11}(t) * [g_1 f(V_1)] + Z_{12}(t) * [g_2 f(V_2)] \\ V_2 &= Z_{21}(t) * [g_1 f(V_1)] + Z_{22}(t) * [g_2 f(V_2)] \\ V_0 &= Z_{01}(t) * [g_1 f(V_1)] + Z_{02}(t) * [g_2 f(V_2)], \end{aligned} \quad (4.16)$$

where V_i ($i = 1, 2$) signifies the membrane voltage in each input region and V_0 the somatic voltage. Voltage 1 and 2 are the sum of two terms: one being the synaptic input current in that region convolved with an input impedance kernel $Z_{ii}(t)$ and the other the synaptic input current in the other region convolved with a transfer impedance kernel $Z_{ij}(t)$, $i \neq j$. The somatic voltage V_0 is also the sum of the synaptic input currents, each convolved with a transfer impedance kernel Z_{0i} from input site to soma. The synaptic current is, as described in § 4.3.2, the product of the synaptic conductance g_i and the SVD $f(V)$.

Each voltage in equation (4.16) depends on both synaptic conductances. This is a general property of the cable equation: the voltage in any region along a neuron depends a priori on all input conductances, whether it is directly, as in the GF approach [Koch, 1998, Wybo et al., 2013], or indirectly through the neighboring voltages, as in compartmental models [Hines, 1984] or the sparse GF approach [Wybo et al., 2015]. Consequently, determining the electrical compartments in neurons has proved to be a daunting task, where one had to resort to extensive simulations and clustering algorithms that do not provide much biological insight [Rabinowitch and Segev, 2006].

Building on a previous proposal that likened the dendritic trees of pyramidal cells to rate-based two-layer neural networks [Poirazi et al., 2003b], we propose a tree graph based approach where the closeness of inputs can immediately be inferred from the topology of the graph (see § 4.4.2 for a full derivation). Hence, predictions can easily be made for any neuron and

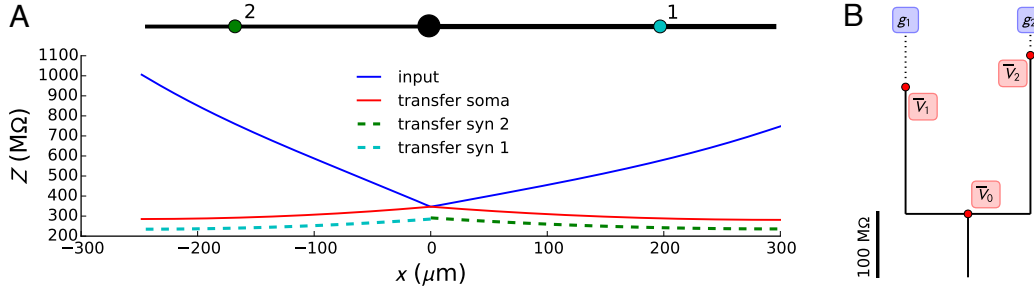


Figure 4.9 – **Neural evaluation tree**. **A:** Impedances on a simple, bipolar neuron. Plotted at position x along the x-axis is the input impedance Z_{xx} in blue, the transfer impedance to the soma Z_{x0} in red, for $x < 0$ the transfer impedance from the left branch to synapse 1 Z_{x1} and, for $x > 0$, the transfer impedance from the right branch to synapse 2 Z_{x2} . **B:** The NET associated with synapses 1 and 2.

any combination of input regions. This tree graph bears resemblance, but is not equivalent to an artificial neural network. It is similar in the sense that, to solve for an output voltage, conductance inputs are first integrated at the leafs of the tree (the first layer) and the solution there is then transferred to their respective parent nodes, which can combine inputs from different leafs. At the root node, an output voltage can be constructed (output layer). This solution algorithm is described in § 4.4.3. When the SVD is simply the driving force, an analytic solution can be derived that bears formal resemblance to artificial neural networks (see § 4.4.4).

We now derive this reformulation for the two-region system (4.16). We start by noting that in many neurons

$$Z_{12} = Z_{21} \approx Z_{01} \approx Z_{02} \quad (4.17)$$

is a reasonable approximation to compute the voltages used to evaluate the SVD's $f(V)$ (note that the equality is a consequence of the symmetry property of the GF). An example is shown in Fig 4.9A: compared to the large input impedances, the transfer impedance to the soma is quite similar to the transfer impedance between the left branch and synapse 1 and between the right branch and synapse 2. Then, by defining a shared root impedance $\bar{Z}_R = Z_{12} = Z_{21}$ and two single-site impedances $\bar{Z}_i = Z_{ii} - Z_{12}$ ($i = 1, 2$), system (4.16) can be rewritten as:

$$\begin{aligned} \bar{V}_1 &= \bar{Z}_1 * \left[g_1 f(\bar{V}_R + \bar{V}_1) \right] \\ \bar{V}_2 &= \bar{Z}_2 * \left[g_2 f(\bar{V}_R + \bar{V}_2) \right] \\ \bar{V}_R &= \bar{Z}_R * \left[g_1 f(\bar{V}_R + \bar{V}_1) + g_2 f(\bar{V}_R + \bar{V}_2) \right] \end{aligned} \quad (4.18)$$

and the location voltage can be obtained as:

$$\begin{aligned} V_1 &= \bar{V}_R + \bar{V}_1 \\ V_2 &= \bar{V}_R + \bar{V}_2 \\ V_0 &\approx \bar{V}_R, \end{aligned} \tag{4.19}$$

where the approximation on the last line is a consequence of assumption (4.17). A schematic of the tree graph associated with this system of equations is shown in Fig 4.9B. Note that the vertical lengths of the branches are proportional to the impedances associated with the nodes. For each input region, the tree has one leaf, representing respectively the variables \bar{V}_1 and \bar{V}_2 . The bottom node of the tree then represents the variable \bar{V}_R that combines both regions. Note furthermore that we have used lines over variables to indicate that they are associated with the NET tree. This is a convention that we will adopt in the rest of this work.

Examining equation (4.18) closely, it can be observed that we have introduced two voltage variables \bar{V}_1 and \bar{V}_2 that feature no direct dependence on the synaptic conductance in the other region. The only dependence on each other is through the root voltage \bar{V}_R in the SVD. Consequently, determining whether regions 1 is independent from 2 becomes a matter only of determining whether the fluctuations induced in \bar{V}_R by synapse 2 are small compared to the fluctuations in \bar{V}_1 induced by synapse 1 and vice versa. In § 4.6 we treat the precise conditions under which a subgroup of input regions can be considered independent from the rest of the input regions.

An added advantage is that the NET system (4.18) is easier to solve than the original system (4.16): in the former there is one less convolution to integrate than in the latter. The reduction in computational cost only increases when more input regions are added, as the number of convolutions in the GF approach scales as the square of the number of input regions [Koch, 1998, Wybo et al., 2013], but the number of nodes does not.

4.4.2 Full neurons: deriving the impedance tree

Due to the symmetry of the transfer impedances, the NET formulation in equation (4.18) is always exact (the only approximation in equation (4.19) is equating the somatic voltage with \bar{V}_R). But how can one derive an approximate NET when there are more than two input regions present on a neuron?

To solve this problem, we order all locations along a dendritic arborization in a depth-first manner [Russell and Norvig, 2003]. Then, the associated impedance matrix becomes highly organized (see as an example Fig 4.10B, where the impedance matrix associated with the granule cell in Fig 4.10A is shown). Generally and with our colour coding, an even blue surface covers most of the matrix. This represents transfer impedances between the main dendritic branches. There are also smaller square regions of light blue or green closer to the diagonal. These regions represent sibling branches, that are thus electrically closer to each other than to

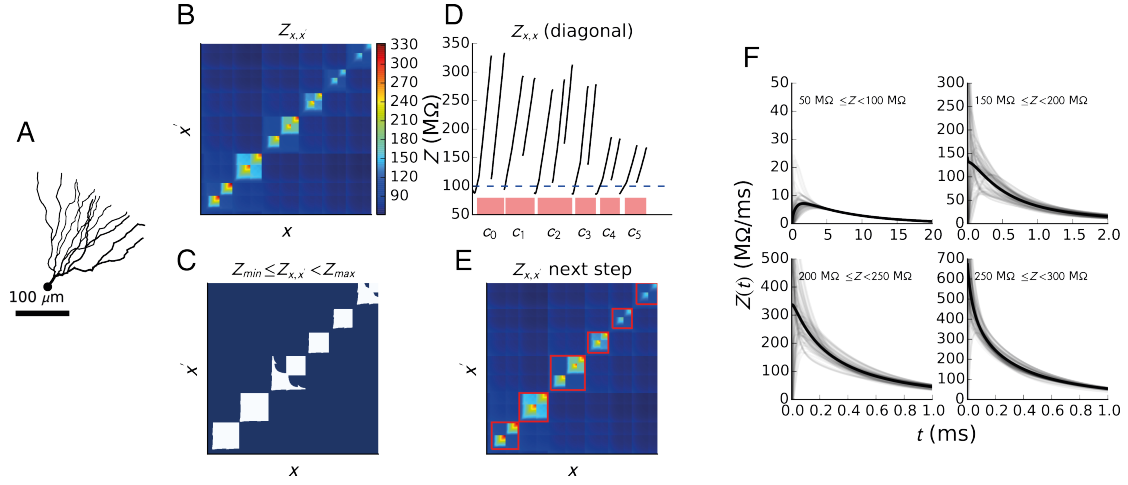


Figure 4.10 – **Illustration of the first step of the recursive algorithm to derive an NET tree.** **A:** A granule cell morphology. **B:** The impedance matrix associated with this cell. **C:** Points where the impedance is between $Z_{\min,0}(= 0\text{M}\Omega)$ and $Z_{\max,0}(= 100\text{M}\Omega)$ are colored blue. **D:** The input impedance (diagonal of the impedance matrix). The connected domains with input impedance larger than $Z_{\max,0}$ are indicated on the x-axis and denoted by c_i , as they will constitute the child nodes of the root in the NET. **E:** The matrices within the red squares correspond to these connected domains. **F:** The average impedance kernels for impedances within the indicated ranges.

different main branches. Finally, the small squares along the diagonal coloured yellow and red are the thin dendritic tips with high input impedances. Consequently, dendritic tips that lie within the same light blue or green square are closer together electrotonically than tips within different squares, as the transfer impedances connecting them are much higher. An NET tree graph structure hence seems to impose itself naturally: the dendritic tips constitute the leaves of the tree, their parent node combines multiple adjacent tips and the root node in turn binds all these nodes together. Note that this impedance matrix is similar to the adjacency matrix proposed in [Cuntz et al., 2010], and, following their conclusions, we hypothesized that it would be sufficient to describe the electrical compartmentalization. Our approach however is more powerful, as we have access to the full temporal profile of the impedance kernels too. In the following we describe a heuristic algorithm to derive NET tree graphs from the impedance matrix that retain all temporal dynamics within the original neuron.

Our algorithm does not assume a discretization in space, as classical compartmental models do, but rather one in impedance. The ‘resolution’ of the NET tree is hence determined by a parameter ΔZ . We employ a recursive algorithm, where each recursion step consist of three phases:

1. Let k denote the step number in our recursive algorithm. We assume that a value $Z_{\min,k}$ is at our disposition. In the first step this value is 0, in later steps it is given by the previous steps. A value $Z_{\max,k}$ is also determined, in the first step as the somatic input

Chapter 4. Dynamic compartmentalization in neurons enables branch-specific learning

impedance (Z_{00} with our depth-first ordering of the impedance matrix) and in later steps by $Z_{\max,k} = Z_{\min,k} + \Delta Z$.

2. The kernel of the current node is constructed as the average of all impedance kernels associated with points in the impedance matrix for which $Z_{\min,k} \leq Z_{ij} < Z_{\max,k}$ (coloured blue in Fig 4.10C). This approach is justified, as impedance kernels of similar magnitude have similar time scales (Fig 4.10F). Then, the kernels of all underlying nodes are subtracted from this average kernel (analogous to how we subtracted Z_{12} from Z_{ii} ($i = 1, 2$) to obtain equation (4.18)).
3. The next nodes are determined by looking at the input impedance, located on the diagonal of the impedance matrix. Due to the depth-first ordering, new nodes can be identified as uninterrupted intervals on this diagonal where $Z_{ii} > Z_{\max,k}$ (Fig 4.10D). For each of these intervals, a new child node is constructed by repeating step 1 with the impedance matrix restricted to the interval (indicated in Fig 4.10E by the red squares) and with $Z_{\min,k+1} = Z_{\max,k}$.

Note that in the NET tree derived in this way, the root node groups all locations along the dendritic arborization together, and, as one moves up through the tree, the nodes group smaller and smaller areas until at the leaf nodes, only the distal tips are integrated.

Generally however, one is only interested in a subset of input regions on the dendrites. In this case the tree can be pruned: leafs that do not integrate an input region of interest can be removed, as can all nodes on the path to the root that do not feature in paths coming from input regions of interest. If multiple nodes integrate the same input region, they can be replaced by one node with an associated impedance kernel that is the sum of all impedance kernels of the original nodes. Furthermore, for each input region on the tree, a leaf node can be added whose impedance kernel fixes the sum of all underlying kernels to be the exact same input impedance kernel as in the original tree.

This algorithm is very successful in constructing NET's of reasonably homogeneous dendritic arborizations. There is one pitfall: since there is only a single 'global' node that integrates all dendritic branches, all transfer impedances across the main branches are the same. This becomes problematic in neurons that have one or multiple very large branches emanating from their cell bodies, which in turn bifurcate more distally in further arborizations, as is for instance the case in pyramidal cells. If all transfer impedances between proximal and distal regions in these cells were the same, interactions between them would be overestimated. In Fig 4.11, it can be seen that such large distal arborizations can be recognized from the histogram of all somatic transfer impedances Z_{0x} (contrast panels A,B in Fig 4.11 with panel C): since such a distal arborization is only connected to the soma through a single branch, there are many points of low Z_{0x} , many of high Z_{0x} (the proximal arborizations), and relatively few of intermediate Z_{0x} . By consequence, the histogram has two main modes. The boundary between these two modes, which we determine using the approach described by [Delon and Desolneux, 2007], then indicates the impedance boundary between proximal and distal domains. Because of

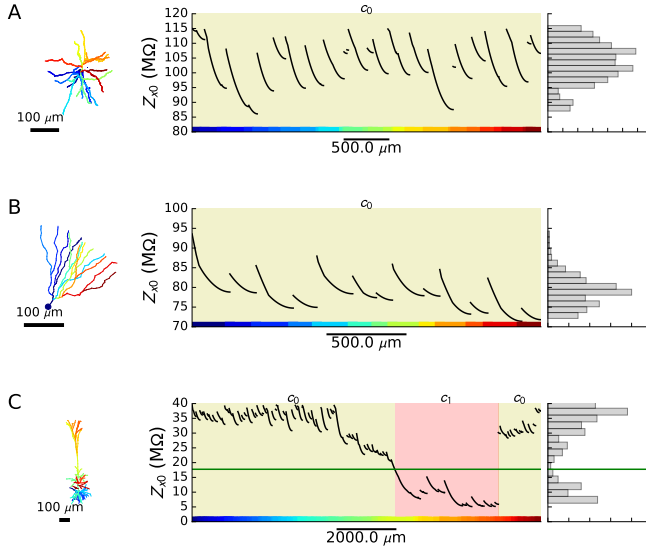


Figure 4.11 – **Distinction between simple and complex dendritic arborizations in the first step of the NET tree generation algorithm.** For each row, the morphology is shown on the left, the bottom row of the impedance matrix (i.e. the somatic transfer impedance $Z_{0,x}$) in the middle, and a histogram of these impedance values on the right. The x-axis of the middle panel is coloured according to the corresponding region on the morphology. The background coloring indicates the different dendritic domains that will become nodes c_i in the NET tree, as found from the histograms on the right. **A:** stellate cell. **B:** granule cell. **C:** pyramidal cell.

the depth first ordering, any region with somatic transfer impedances above this boundary will belong to the proximal domain (its associated node is denoted by c_0 in Fig 4.11), whereas connected regions below this boundary will constitute the distal domains (coloured red in Fig 4.11C, and with c_1 its associated node).

Algorithmically, our approach is thus the following: first, we determine the kernel of the root node as the average of all transfer impedance kernels between proximal and distal branches. Then, we start the recursive procedure as before, but with the impedance matrices restricted to the different domains c_i .

The final question left to ask is what an appropriate choice for ΔZ would be. As the NET tree is an approximation, even an infinitesimally small ΔZ will not yield an error in the impedance matrix of zero. Examining Fig 4.12, it can be observed that the decrease in error is negligible once ΔZ drops below 10 to 20 $\text{M}\Omega$. We consequently choose values within that range and obtained accurate reproductions of voltage traces computed with the NEURON simulator [Carnevale and Hines, 2006].

4.4.3 An efficient solution algorithm

On the one hand, the NET allows efficient simulation of the neuronal dynamics, and on the other hand it can be used to compute attractor points of these dynamics. The latter yield insight into the asymptotic behaviour of the neuron as a dynamical system and can be used

to extract information about compartmentalization: for an instantaneous set of synaptic conductances $g_1(t), \dots, g_n(t)$, the instantaneous solution of the attractor point will be what the full dynamical system is trying to reach. Hence, if components of the attractor point solution behave independently from other components, the full dynamics will behave independently too. Both the full simulation and attractor point solution involve solving linear systems by matrix inversion. The NET tree graph structure allows this to be done in $O(n)$ steps, as opposed to $O(n^3)$ steps for arbitrary linear systems. We will first describe how the nonzero elements in this matrix are constructed for each solution method. Then we will describe the solution algorithm.

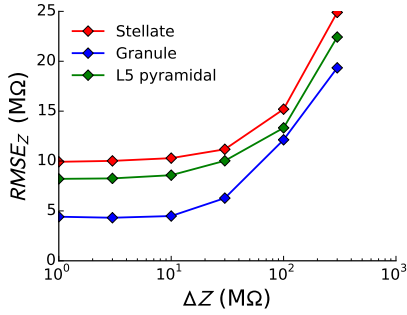


Figure 4.12 – **Error of the NET approximation.** The root mean square error of the impedance matrix approximated by the NET tree compared to the true impedance matrix, as a function of ΔZ .

We will order the node voltages in the NET tree in a reversed depth-first manner (so that the root voltage comes last) and group them in a vector $\bar{\mathbf{V}}$. Similarly, the node impedances will be grouped in a vector $\bar{\mathbf{Z}}$. With each node N we will associate the set \mathcal{R}_N containing all input regions that a node integrates. Furthermore, with each input region r we will associate the set \mathcal{S}_r of all synapse types present in the input region and the set \mathcal{N}_r of all nodes that integrate inputs from that region. Although not strictly necessary, we assume for simplicity that each leaf only integrates inputs from one region and that each input region is first integrated by a leaf. With these assumptions, the sets \mathcal{N}_r always constitute paths from leaf to root, as is illustrated in Fig 4.13A. The general form of the synaptic input current at node N (the factor in square brackets in (4.18)) is then given by:

$$\bar{I}_N(t, \bar{\mathbf{V}}(t)) = \sum_{r \in \mathcal{R}_N} \sum_{s \in \mathcal{S}_r} g_s(t) f_s \left(\sum_{M \in \mathcal{N}_r} \bar{V}_M(t) \right). \quad (4.20)$$

We thus sum over all regions integrated by node N and all synapse types in these input regions, and the voltage in the SVD is given by the sum over all node voltages that integrate these input regions. Note that besides $\bar{\mathbf{V}}(t)$, the only other temporal dependence in $\bar{\mathbf{I}}(t, \bar{\mathbf{V}}(t))$ is through the synaptic conductances ($g_s(t)$, grouped in a vector \mathbf{g}). When convenient, we will therefor sometimes write $\bar{\mathbf{I}}_{\mathbf{g}(t)}(\bar{\mathbf{V}}(t))$. Finally, we remark that the set \mathcal{N}_r always denotes a direct path from leaf to root.

The full dynamical system. The full dynamical system associated with the NET can be written in vector form as:

$$\bar{\mathbf{V}}(t) = \int_{-\infty}^t ds \bar{\mathbf{Z}}(t-s) \odot \bar{\mathbf{I}}(s, \bar{\mathbf{V}}(s)). \quad (4.21)$$

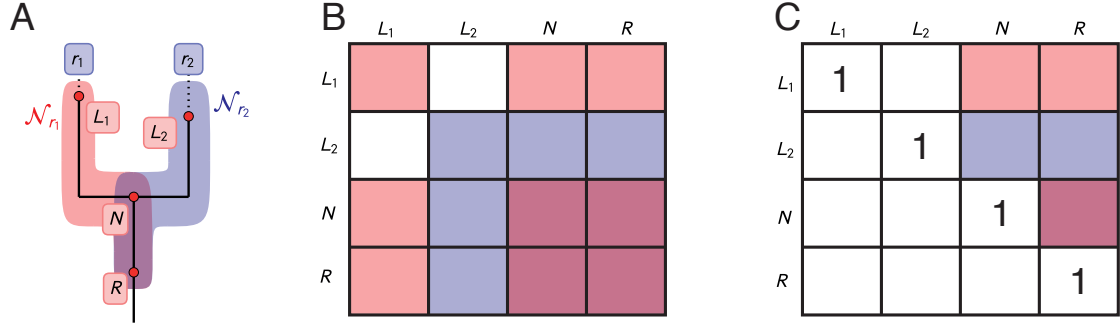


Figure 4.13 – **Schematic representation of the inversion algorithm.** **A:** A schematic of a simple NET with four nodes L_1, L_2, N and R , two input regions r_1 and r_2 , and the two paths \mathcal{N}_{r_1} and \mathcal{N}_{r_2} coloured red and blue respectively (mixed colour purple indicates that the two paths overlap). **B:** The structure of the associated matrix \mathbb{A} as obtained from equation (4.32). **C:** By a recursive series of row operations the sub-diagonal elements of \mathbb{A} are set to zero, while the diagonal elements are at the same time set to one. After the down sweep, the root voltage can be computed directly. Other voltages are then computed during the up sweep.

where \odot denotes element-wise multiplication. To solve a system of this form for $\bar{\mathbf{V}}(t+h)$ when $\bar{\mathbf{I}}(t-kh, \bar{\mathbf{V}}(t-kh))$ is known for $k=0, 1, 2, \dots$, one can assume a linear interpolation between the grid points. Since the impedances $\bar{Z}_N(t)$ are known as sums of exponentials, they can be integrated analytically in combination with the linear interpolation to obtain an accurate quadrature rule (section 2.2 in [Wybo et al., 2015]). Grouping all terms that do not contain $\bar{\mathbf{V}}(t+h)$ in a term $\bar{\mathbf{F}}(t)$, one obtains:

$$\bar{\mathbf{V}}(t+h) = \bar{\mathbf{Z}}^{(0)} \odot \bar{\mathbf{I}}(t+h, \bar{\mathbf{V}}(t+h)) + \bar{\mathbf{F}}(t), \quad (4.22)$$

where $\bar{\mathbf{Z}}^{(0)}$ is a vector based on the impedance kernels that follows from the quadrature rule. Taylor expanding \bar{I}_N around $\bar{\mathbf{V}}(t)$ gives:

$$\bar{I}_N(t+h, \bar{\mathbf{V}}(t+h)) \approx \bar{I}_N(t+h, \bar{\mathbf{V}}(t)) + \sum_M \frac{\partial \bar{I}_N}{\partial \bar{V}_M} \Big|_{t+h, \bar{\mathbf{V}}(t)} [\bar{V}_M(t+h) - \bar{V}_M(t)]. \quad (4.23)$$

Moving all terms containing components of $\bar{\mathbf{V}}(t+h)$ to the left hand side then gives a system of the form

$$\mathbb{A}(t+h) \bar{\mathbf{V}}(t+h) = \bar{\mathbf{B}}(t+h), \quad (4.24)$$

which can be solved for the node voltage. In this system, the vector $\bar{\mathbf{B}}(t+h)$ contains:

$$\bar{B}_N(t+h) = \bar{Z}_N^{(0)} \left[\bar{I}_N(t+h, \bar{\mathbf{V}}(t)) - \sum_M \frac{\partial \bar{I}_N}{\partial \bar{V}_M} \Big|_{t+h, \bar{\mathbf{V}}(t)} \bar{V}_M(t) \right] + \bar{F}_N(t) \quad (4.25)$$

and the elements of matrix $\mathbb{A}(t+h)$ are given by:

$$A_{NM}(t+h) = \delta_{NM} - \bar{Z}_N^{(0)} \frac{\partial \bar{I}_N}{\partial \bar{V}_M} \Big|_{t+h, \bar{\mathbf{V}}(t)}, \quad (4.26)$$

with δ_{NM} the Kronecker delta.

The attractor points. The attractor points associated with (4.21) follow from the equality:

$$\bar{\mathbf{V}} = \bar{\mathbf{Z}} \odot \bar{\mathbf{I}}_{\mathbf{g}}(\bar{\mathbf{V}}) \quad (4.27)$$

or equivalently:

$$\bar{\mathbf{H}}(\bar{\mathbf{V}}) := \bar{\mathbf{V}} - \bar{\mathbf{Z}} \odot \bar{\mathbf{I}}_{\mathbf{g}}(\bar{\mathbf{V}}) = 0 \quad (4.28)$$

and can be found from the Newton iteration (with k the iteration index):

$$\mathbb{A}_{\bar{\mathbf{H}}(\bar{\mathbf{V}}_k)} \bar{\mathbf{V}}_{k+1} = \bar{\mathbf{B}}_k, \quad (4.29)$$

with $\mathbb{A}_{\bar{\mathbf{H}}(\cdot)}$ the Jacobian of $\bar{\mathbf{H}}(\cdot)$ and the right-hand side of this equation given by:

$$\bar{\mathbf{B}}_k = -\bar{\mathbf{H}}(\bar{\mathbf{V}}_k) + \mathbb{A}_{\bar{\mathbf{H}}(\bar{\mathbf{V}}_k)} \bar{\mathbf{V}}_k. \quad (4.30)$$

The elements of the Jacobian are:

$$A_{NM} = \delta_{NM} - \bar{Z}_N \frac{\partial \bar{I}_{\mathbf{g},N}}{\partial \bar{V}_M}(\bar{\mathbf{V}}_k). \quad (4.31)$$

The matrix inversion. From (4.26) and (4.31) it can be seen that systems (4.24) and (4.29) have the same structure, the only difference being that in (4.26) constants $\bar{Z}_N^{(0)}$ are employed that follow from the quadrature rule whereas in (4.31) the node impedances \bar{Z}_N are used. The NET tree now imposes a very special structure on the matrix \mathbb{A} that allows it to be inverted in $O(n)$ steps. First, the algorithm performs a single down sweep that sets all elements under the diagonal to zero. Then, a single up sweep is performed to set all elements above the diagonal to zero. To construct \mathbb{A} , we start from the identity matrix and then subtract

$$\bar{Z}_N \frac{\partial \bar{I}_{\mathbf{g},N}}{\partial \bar{V}_M} = \bar{Z}_N \sum_{r \in \mathcal{R}_N} \left[\sum_{s \in \mathcal{S}_r} \left(g_s f'_s \left(\sum_{K \in \mathcal{N}_r} \bar{V}_K \right) \right) \sum_{K \in \mathcal{N}_r} \delta_{KM} \right] \quad (4.32)$$

for all N, M . Here, f'_s denotes the derivative of f_s . Dissecting this, we see that for each path \mathcal{N}_r , the rows corresponding to the nodes $N \in \mathcal{N}_r$ contain a term $-\bar{Z}_N \left[\sum_{s \in \mathcal{S}_r} g_s f'_s \left(\sum_{K \in \mathcal{N}_r} \bar{V}_K \right) \right]$ at each column associated with a node in \mathcal{N}_r . This leads to a structure that is depicted in

Fig 4.13B. To simplify the notation, we will denote the term in square brackets by:

$$a(\mathcal{N}_r) := \sum_{s \in \mathcal{S}_r} g_s f'_s \left(\sum_{K \in \mathcal{N}_r} \bar{V}_K \right). \quad (4.33)$$

We will also group all terms in columns associated with \mathcal{N}_r in row N of the matrix in a vector $\mathbf{R}(N)|_{\mathcal{N}_r}$, that thus contains

$$\left(\mathbf{R}(N)|_{\mathcal{N}_r} \right)_K = \delta_{NK} - \bar{Z}_N a(\mathcal{N}_r), \quad \text{for } K \in \mathcal{N}_r. \quad (4.34)$$

For instance, in Fig 4.13B, $\mathbf{R}(L_1)|_{\mathcal{N}_{r_1}}$ is given by the red elements on the first line of the matrix, and similarly for nodes N and R . In turn, rows $\mathbf{R}(\cdot)|_{\mathcal{N}_{r_2}}$ associated with \mathcal{N}_{r_2} are coloured blue for nodes L_2, N and R .

Finally, we introduce for formal reasons a term $b(\mathcal{N}_r)$ that is zero in the first step of the algorithm, so that the elements of the vector \mathbf{B} become:

$$(\bar{\mathbf{B}})_N = \bar{B}_N - \bar{Z}_N \sum_{r \in \mathcal{R}_N} b(\mathcal{N}_r), \quad (4.35)$$

and similarly to $\mathbf{R}(N)|_{\mathcal{N}_r}$, we will take $(\bar{\mathbf{B}}|_{\mathcal{N}_r})$ to signify the restriction of $\bar{\mathbf{B}}$ to terms only in \mathcal{N}_r :

$$\left(\bar{\mathbf{B}}|_{\mathcal{N}_r} \right)_N = \bar{B}_N - \bar{Z}_N b(\mathcal{N}_r). \quad (4.36)$$

First, we describe the steps of the down sweep phase of the algorithm.

1. We take the leaf L of a path \mathcal{N}_r , divide row L by its diagonal element and obtain:

$$\left(\mathbf{R}(L)|_{\mathcal{N}_r} \right)_K = \begin{cases} \frac{-\bar{Z}_L a(\mathcal{N}_r)}{1 - \bar{Z}_L a(\mathcal{N}_r)} & \text{if } K \neq L \\ 1 & \text{if } K = L, \end{cases} \quad (4.37)$$

Correspondingly,

$$(\bar{\mathbf{B}})_L = \frac{\bar{B}_L - \bar{Z}_L b(\mathcal{N}_r)}{1 - \bar{Z}_L a(\mathcal{N}_r)}. \quad (4.38)$$

On all rows $N \in \mathcal{N}_r \setminus \{L\}$ we effectuate the row operation

$$\begin{aligned} \mathbf{R}(N) &\rightarrow \mathbf{R}(N) - \left(\mathbf{R}(N)|_{\mathcal{N}_r} \right)_L \mathbf{R}(L) \\ (\bar{\mathbf{B}})_N &\rightarrow (\bar{\mathbf{B}})_N - \left(\mathbf{R}(N)|_{\mathcal{N}_r} \right)_L (\bar{\mathbf{B}})_L. \end{aligned} \quad (4.39)$$

This operation sets sub-diagonal elements $\left(\mathbf{R}(N)|_{\mathcal{N}_r} \right)_L$ to zero. For the non-zero ele-

ments $M \neq L$ we obtain:

$$\begin{aligned} (\mathbf{R}(N)|_{\mathcal{N}_r})_M &= \delta_{NM} - \bar{Z}_N a(\mathcal{N}_r) - \left(-\bar{Z}_N a(\mathcal{N}_r) \right) \frac{-\bar{Z}_L a(\mathcal{N}_r)}{1 - \bar{Z}_L a(\mathcal{N}_r)} \\ &= \delta_{NM} - \bar{Z}_N \frac{a(\mathcal{N}_r)}{1 - \bar{Z}_L a(\mathcal{N}_r)}, \end{aligned} \quad (4.40)$$

while

$$\begin{aligned} (\bar{\mathbf{B}}|_{\mathcal{N}_r})_N &= \bar{B}_N - \bar{Z}_N b(\mathcal{N}_r) - \left(-\bar{Z}_N a(\mathcal{N}_r) \right) \frac{\bar{B}_L - \bar{Z}_L b(\mathcal{N}_r)}{1 - \bar{Z}_L a(\mathcal{N}_r)} \\ &= \bar{B}_N - \bar{Z}_N \frac{b(\mathcal{N}_r) - \bar{B}_L a(\mathcal{N}_r)}{1 - \bar{Z}_L a(\mathcal{N}_r)}. \end{aligned} \quad (4.41)$$

Let P be the parent node of L in the path \mathcal{N}_r . If all sub-diagonal elements for paths associated with regions in \mathcal{R}_P have been set to zero, one can move on to step 2 with node P . Otherwise repeat step 1 with a different leaf and path until such is the case.

2. Consider all $r \in \mathcal{R}_P$. We will denote the restriction of the path \mathcal{N}_r to nodes from P to the root as $\mathcal{N}_r^{P\downarrow}$. The tree structure imposes that all these paths are the same:

$$\mathcal{N}_{r_1}^{P\downarrow} = \mathcal{N}_{r_2}^{P\downarrow}, \quad \forall r_1, r_2 \in \mathcal{R}_P. \quad (4.42)$$

Hence we will use the shorthand:

$$\mathcal{N}_P := \mathcal{N}_r^{P\downarrow}. \quad (4.43)$$

As a consequence of step 1, it holds for all $N \in \mathcal{N}_P$ that:

$$(\mathbf{R}(N)|_{\mathcal{N}_P})_M = \delta_{NM} - \bar{Z}_N \sum_{r \in \mathcal{R}_P} \frac{a(\mathcal{N}_r)}{1 - \bar{Z}_{L_r} a(\mathcal{N}_r)}, \quad (4.44)$$

where L_r signifies the leaf associated with the path \mathcal{N}_r , and that:

$$(\bar{\mathbf{B}}|_{\mathcal{N}_P})_N = \bar{B}_N - \bar{Z}_N \sum_{r \in \mathcal{R}_P} \frac{b(\mathcal{N}_r) - \bar{B}_{L_r} a(\mathcal{N}_r)}{1 - \bar{Z}_{L_r} a(\mathcal{N}_r)}. \quad (4.45)$$

We can now define

$$a(\mathcal{N}_P) := \sum_{r \in \mathcal{R}_P} \frac{a(\mathcal{N}_r)}{1 - \bar{Z}_{L_r} a(\mathcal{N}_r)} \quad (4.46)$$

and

$$b(\mathcal{N}_P) := \sum_{r \in \mathcal{R}_P} \frac{b(\mathcal{N}_r) - \bar{B}_{L_r} a(\mathcal{N}_r)}{1 - \bar{Z}_{L_r} a(\mathcal{N}_r)}. \quad (4.47)$$

With these substitutions, we are now able to formally proceed with the algorithm as if P

was a leaf and \mathcal{N}_P the path to it. We may thus return to step 1, unless P was the root of the tree, in which case its associated voltage is known:

$$\bar{V}_P = \frac{\bar{B}_P - \bar{Z}_P b(\mathcal{N}_P)}{1 - \bar{Z}_P a(\mathcal{N}_P)}. \quad (4.48)$$

The structure of the matrix after the down sweep is illustrated in Fig 4.13C.

Note that the efficiency of the algorithm lies in the fact that we do not have to do the row operation (4.39) explicitly for each node N on the path to the root. It suffices to compute the quantities $a(\mathcal{N}_P)$ and $b(\mathcal{N}_P)$ once for each node that is not a leaf.

For the up sweep of the algorithm, we assume that we have a node P for which the voltages associated with nodes in \mathcal{N}_P (i.e. the path from P down to the root) are known. Then performing, for all nodes N in the sub-tree of P , the row operation:

$$\begin{aligned} \mathbf{R}(N) &\rightarrow \mathbf{R}(N) - (\mathbf{R}(N))_P \mathbf{R}(P) \\ (\bar{\mathbf{B}})_N &\rightarrow (\bar{\mathbf{B}})_N - (\mathbf{R}(N))_P (\bar{\mathbf{B}})_P \end{aligned} \quad (4.49)$$

sets the supra-diagonal elements $(\mathbf{R}(N)|_{\mathcal{N}_n})_P$ to zero. Repeating this operation with the child nodes C of P sets their associated supra-diagonal elements $(\mathbf{R}(N)|_{\mathcal{N}_n})_C$ to zero, with N now in the sub-tree of C . Continuing so until the leaves are reached diagonalizes the whole matrix.

Key here is that we do not have to do all the row operations explicitly either; from equation (4.44) it follows that all supra-diagonal elements on a given row are the same. Hence, the voltage associated with a node N with parent P is given by:

$$\bar{V}_N = \frac{\bar{B}_N - \bar{Z}_N b(\mathcal{N}_n)}{1 - \bar{Z}_N a(\mathcal{N}_n)} + \frac{\bar{Z}_N a(\mathcal{N}_n)}{1 - \bar{Z}_N a(\mathcal{N}_n)} \left(\sum_{K \in \mathcal{N}_P} \bar{V}_K \right) \quad (4.50)$$

and recursion can proceed with the child nodes of N .

To conclude, we summarize the algorithm in the pseudocode in Fig 4.14.

4.4.4 Conductance-based synapses: analytic solutions

The terms in $\bar{\mathbf{I}}_g(\bar{\mathbf{V}})$ corresponding to AMPA and GABA synapses depend linearly on the voltage through the driving force. When only such synapses are present, the Newton iteration (4.29) yields the true node voltages in a single step, while at the same time providing insight in the input/output relation of the neuron. We describe this input/output relation in the next paragraph for the case where only a single synapse type is present, although the extension to multiple synapse types is possible. Furthermore, we show that in certain cases, a shunting input can be seen as reducing the impedance associated with specific nodes in the NET tree. We also discuss the relation with the shunt level, a quantity used previously to quantify

<p>Procedure Down sweep(<i>leaf</i> L)</p> <pre> 1 if L is true leaf then 2 compute $a(\mathcal{N}_l)$ 3 $b(\mathcal{N}_l) \leftarrow 0$ end 4 compute B_L according to integration paradigm 5 store $B_L, a(\mathcal{N}_l), b(\mathcal{N}_l)$ at node L 6 if L not root then 7 retrieve parent P 8 $a(\mathcal{N}_P) += \frac{a(\mathcal{N}_l)}{1 - \bar{Z}_L a(\mathcal{N}_l)}$ 9 $b(\mathcal{N}_P) += \frac{b(\mathcal{N}_l) - \bar{B}_L a(\mathcal{N}_l)}{1 - \bar{Z}_L a(\mathcal{N}_l)}$ end 10 if all children of P have been passed then 11 if P is not root then 12 $L \leftarrow P$ 13 Down sweep (L) end 14 else 15 Stop recursion end end 16 else 17 $L \leftarrow$ next leaf 18 Down sweep (L) end </pre> <p>Input: leaf L Down sweep(L)</p>	<p>Procedure Up sweep(<i>node</i> N, voltage V_P)</p> <pre> 1 $\bar{V}_N \leftarrow \frac{\bar{B}_N - \bar{Z}_N b(\mathcal{N}_n)}{1 - \bar{Z}_N a(\mathcal{N}_n)} + \frac{\bar{Z}_N a(\mathcal{N}_n)}{1 - \bar{Z}_N a(\mathcal{N}_n)} V_P$ 2 $a(\mathcal{N}_n) \leftarrow 0, b(\mathcal{N}_n) \leftarrow 0, B_N \leftarrow 0$ 3 for childnodes C of N do 4 Up sweep($C, V_P + \bar{V}_N$) end </pre> <p>Input: root R Up sweep($R, 0$)</p>
---	--

Figure 4.14 – Pseudocode for the up and down sweep phases of the matrix inversion algorithm. During both phases, each node is passed exactly once.

shunting interactions in dendritic trees [Gidon and Segev, 2012].

A simple input/output transformation. When a single synapse type with linear voltage dependence is present on the dendritic tree, equation (4.20) becomes:

$$\bar{I}_{N,\mathbf{g}}(\bar{\mathbf{V}}) = \sum_{r \in \mathcal{R}_N} g_r \left(E_r - V_{\text{eq}} - \sum_{K \in \mathcal{N}_r} \bar{V}_K \right), \quad (4.51)$$

where E_r denotes the synaptic reversal potential and V_{eq} the equilibrium voltage. For the sake of brevity, we will group these time-independent potentials in a single variable :

$$E := E_r - V_{\text{eq}} \quad (4.52)$$

The elements of matrix \mathbb{A} in (4.29) have the form:

$$A_{NM} = \delta_{NM} + \bar{Z}_N \sum_{r \in \mathcal{R}_N} g_r \sum_{K \in \mathcal{N}_r} \delta_{KM} \quad (4.53)$$

whereas for the contents of vector $\bar{\mathbf{B}}$ (with $\bar{\mathbf{V}}_0 = 0$) we find:

$$\bar{B}_N = Z_N \sum_{r \in \mathcal{R}_N} g_r E. \quad (4.54)$$

For the rows of \mathbb{A} , this means that:

$$\left(\mathbf{R}(N) \big|_{\mathcal{N}_r} \right)_K = \delta_{NK} + \bar{Z}_N g_r, \quad \text{for } K \in \mathcal{N}_r. \quad (4.55)$$

In step 2 of the down sweep algorithm, with these simplifications it holds that:

$$\left(\mathbf{R}(N) \big|_{\mathcal{N}_p} \right)_M = \delta_{NM} + Z_N \sum_{r \in \mathcal{R}_p} \frac{g_r}{1 + Z_{L_r} g_r} \quad (4.56)$$

and

$$\begin{aligned} \left(\bar{\mathbf{B}} \big|_{\mathcal{N}_r} \right)_N &= \bar{B}_N - \sum_{r \in \mathcal{V}_p} \bar{Z}_N g_r \frac{Z_{L_r} g_r E}{1 + Z_{L_r} g_r} \\ &= \bar{Z}_N \sum_{r \in \mathcal{R}_p} g_r E - \sum_{r \in \mathcal{R}_p} \bar{Z}_N g_r \frac{Z_{L_r} g_r E}{1 + Z_{L_r} g_r} \\ &= \bar{Z}_N \sum_{r \in \mathcal{R}_p} \frac{g_r}{1 + \bar{Z}_{L_r} g_r} E. \end{aligned} \quad (4.57)$$

This suggest that instead of equation (4.46), we can now define:

$$g_p := \sum_{r \in \mathcal{R}_p} \frac{g_r}{1 + \bar{Z}_{L_r} g_r}, \quad (4.58)$$

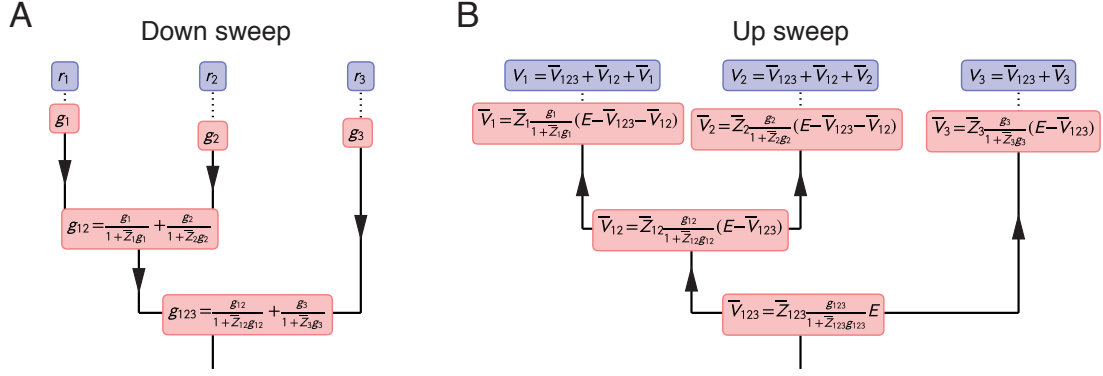


Figure 4.15 – **Solution for a single synapse type.** **A:** In the down sweep phase, the conductances are rescaled by a shunt factor. **B:** The up sweep sets all the node voltages.

and, since this factor occurs both in (4.56) and (4.57), we may consider it as a synaptic conductance associated with node P . With respect to the downward recursion, P then formally becomes a leaf of the tree. If P is the root of the tree, its voltage is given by:

$$\bar{V}_P = \bar{Z}_P \frac{g^P}{1 + \bar{Z}_P g^P} E. \quad (4.59)$$

The up sweep then proceeds as in the full algorithm, but instead of equation (4.50), we find:

$$\bar{V}_N = \bar{Z}_N \frac{g^N}{1 + \bar{Z}_N g^N} \left(E - \sum_{K \in \mathcal{N}_P} \bar{V}_K \right). \quad (4.60)$$

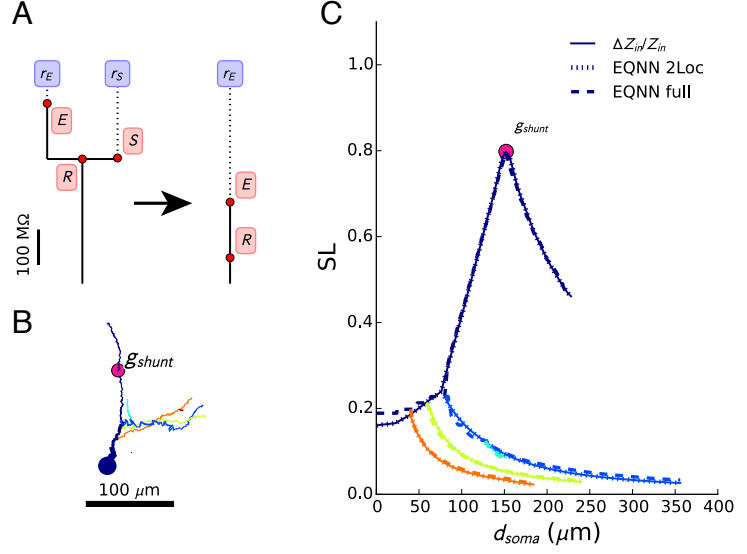
This solution is depicted in Fig 4.15 for a simple NET tree. It can be seen that the conductance associated with the root node is given by a combination of shunting non-linearities

$$F_Z(g) = \frac{g}{1 + Zg} \quad (4.61)$$

and linear sums (Fig 4.15A), and that the root voltage can be computed directly from knowledge of the associated conductance. This parallels the way the output is constructed in artificial neural networks (ANN's). The amount of shunting in a node is furthermore proportional to the impedance in that node. If conductance and/or impedance are small, so that $Zg \ll 1$ in all nodes, the root voltage is simply a linear sum of all the inputs. If local voltages need to be known, the up sweep is straightforward to compute, as depicted in Fig 4.15B.

Shunting. Shunting inhibition has proven to be a powerful tool to modulate the output of regions in the dendritic arborization [Gidon and Segev, 2012]. The influence of this shunt on other dendritic regions can be quantified by the shunt level SL , i.e. the relative reduction in

Figure 4.16 – *SL in a subtree of a stellate cell*. **A:** Schematic NET with a region of excitation r_E and a region of shunting r_S . The shunt can be eliminated from the NET by rescaling the conductance of the root R . **B:** The dendritic subtree. **C:** *SL* computed according to the normal definition (labeled $\Delta Z_{in}/Z_{in}$), the exact two location NET (labeled 2Loc) and an approximate NET pruned from the full NET tree (labeled full). Colors correspond to the colour code in B.



input impedance:

$$SL = \frac{\Delta Z_{in}}{Z_{in}}. \quad (4.62)$$

It is instructive to understand the analytical NET formula for this shunt level. The original NET system has the following form:

$$\begin{aligned} \bar{V}_E &= \bar{Z}_E g_E \cdot f(\bar{V}_R + \bar{V}_E) \\ \bar{V}_S &= -\bar{Z}_S g_S \cdot (\bar{V}_R + \bar{V}_S) \\ \bar{V}_R &= \bar{Z}_R \left[g_E \cdot f(\bar{V}_R + \bar{V}_E) - g_S \cdot (\bar{V}_R + \bar{V}_S) \right], \end{aligned} \quad (4.63)$$

where the subscript E denotes a node associated with only excitation, S a node with only shunting inhibition and R the root node that combines both synapses. Solving for shunt-related terms gives:

$$\begin{aligned} \bar{V}_E &= \bar{Z}_E g_E \cdot f(\bar{V}_R + \bar{V}_E) \\ \bar{V}_S &= -\frac{\bar{Z}_S}{1 + \bar{Z}_S g_S} g_S \bar{V}_R \\ \bar{V}_R &= \frac{\bar{Z}_R}{1 + \bar{Z}_R \frac{g_S}{1 + \bar{Z}_S g_S}} \left[g_E \cdot f(\bar{V}_R + \bar{V}_E) \right]. \end{aligned} \quad (4.64)$$

Formally, the first and last equations in this system can now be seen as an NET with a single input region and a root impedance that is modified due to the shunt conductance g_S , as is illustrated in Fig 4.16A. From the original input impedance at the excitatory synapse $\bar{Z}_E + \bar{Z}_R$

and the new input impedance $\bar{Z}_E + \frac{\bar{Z}_R}{1 + \bar{Z}_R \frac{g_S}{1 + \bar{Z}_S g_S}}$, the shunt level can be computed:

$$SL = \frac{\overbrace{\bar{Z}_R g_S}^1}{\underbrace{\left(1 + \frac{\bar{Z}_E}{\bar{Z}_R}\right)}_2 \underbrace{\left(1 + (\bar{Z}_R + \bar{Z}_S)g_S\right)}_3}. \quad (4.65)$$

It can be seen from factor 1 in the above formula that SL is proportional to the impedance in the common node as well as the shunt conductance. However, SL can not grow indeterminately, due to the saturation factor 3. The maximal shunt level is thus inversely proportional to the input impedance at the shunting synapse, $\bar{Z}_R + \bar{Z}_S$. Finally, due to factor 2, the amount of shunting also depends on the ratio between impedance at the excitatory node versus impedance at the root node. We computed SL for a subtree of a stellate cell (Fig 4.16B), once using the definition (4.62), and once using the NET derived for two locations ('NET 2loc' in Fig 4.16C). Since the NET for two locations is exact (see § 4.4.1), these two lines coincide perfectly. We also computed SL once for an NET tree obtained from the algorithm described in § 4.4.2 by pruning all unneeded regions. This approximate tree agrees very well with the true SL , thus serving as a further validation that our full NET tree can capture the interactions within a dendritic arborization accurately.

4.5 Predicting spikes: linear layers

While our NET approximation is sufficiently accurate in computing the voltages for the SVD's, the very sharp threshold for AP generation at the soma still requires a higher precision. This precision is hard to achieve with the NET as it is, since the effective transfer impedance between the soma and the dendritic loci is either the same everywhere, or can take on only a proximal and distal value (see § 4.4.2). We mitigate this problem by introducing 'linear layers' (LinL's).

In our NET implementation, there are no constraints on the channel types that can be included in the AP current. Furthermore, the associated currents are modeled in the same way as the synaptic current. AP channels contribute to the nodal input current only at nodes that integrate the somatic region. Usually, there are two such nodes: the root of the NET tree R and a leaf L_0 . The latter represents a component of the voltage only felt in and around the soma. The AP current contribution at these nodes to \bar{I}_N ($N = R, L_0$) can be written as $I_{AP}(V_R + V_{L_0})$. The LinL's now add an extra term V_{lin} to the voltage this current depends on. This term has the form:

$$V_{lin} = \sum_{l \in \mathcal{L}} \bar{Z}_l(t) * \sum_{r \in \mathcal{R}_l} \left[\sum_{s \in \mathcal{S}_r} g_s f\left(\sum_{M \in \mathcal{N}_r} \bar{V}_M\right) \right], \quad (4.66)$$

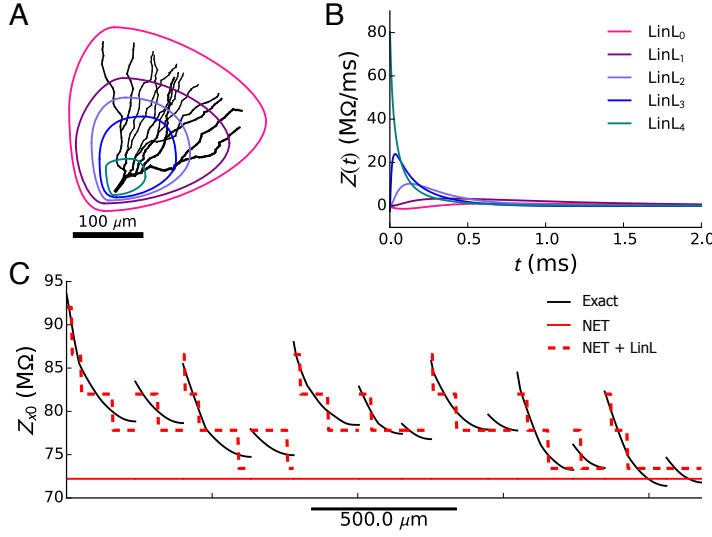


Figure 4.17 – **LinL's to improve dendrite to soma transfer.** **A:** Schematic of the spatial extent of the five different LinL's. **B:** The impedance kernels associated with each LinL (color coding corresponds to A). **C:** The transfer impedance from input site to soma (first column or bottom row in the impedance matrix, shown in Fig 4.11). Shown is the exact value (black), the NET value (red, full line) and the NET + LinL value (red, dashed line).

where \mathcal{L} is the set of LinL's, \mathcal{R}_l is the set of regions a linear layer integrates and $\bar{Z}_l(t)$ is the impedance kernel associated with a layer l . Since this term is only felt by I_{AP} , it does not contribute to the SVD's.

In Fig 4.17 the LinL's are shown for the same granule cell as in Fig 4.10. The sets \mathcal{R}_L are illustrated in Fig 4.17A. As can be seen from Fig 4.17B, the LinL with the largest spatial extent, i.e. the one that integrates all regions, has the smallest kernel. LinL's that only integrate more proximal inputs then add shorter-time scale kernels to this. These LinL's hence serve to correctly approximate delay and attenuation of the dendritic inputs. In Fig 4.17C, the transfer impedance to the soma (which corresponds to both the leftmost column and the bottom row in the impedance matrix $Z_{xx'}$) is plotted. It can be seen that the NET in combination with the LinL's constitutes a much more faithful approximation than without. We also note that the maximal impedance difference that is implemented by the LinL's is around 20 MΩ. This is small compared to the local impedance that rises to over 300 MΩ in this cell (Fig 4.10B). In many cortical cells, this difference is even larger, as local impedances often go up to 1000 or even 2000 MΩ, whereas the difference in somatic transfer impedance is also around 20 to 50 MΩ.

4.5.1 Matrix inversion with linear layers.

To extend the matrix inversion algorithm explained in § 4.4.3, we note that the rows in the Jacobian associated with nodes that integrate the AP current receive extra terms. For each region r , and for all nodes K on the path \mathcal{N}_r , the following terms are added to rows associated

with all nodes N integrating the AP current:

$$\left(\mathbf{R}(N)|_{\mathcal{N}_r}\right)_K^{\text{LinL}} = -\bar{Z}_N I'_{AP} \sum_{l \in \mathcal{L}_r} \bar{Z}_l a(\mathcal{N}_r), \quad (4.67)$$

where $\mathcal{L}_r = \{l \in \mathcal{L} | r \in l\}$ is the set of all LinL's that receive input from region r . For ease of notation we will define a quantity α_r :

$$\alpha_r = \sum_{l \in \mathcal{L}_r} \bar{Z}_l a(\mathcal{N}_r), \quad (4.68)$$

as well as a term β_r that is zero in the first step of the down sweep algorithm,

$$\left(\bar{\mathbf{B}}|_{\mathcal{N}_r}\right)_N^{\text{LinL}} = -\bar{Z}_N I'_{AP} \beta_r. \quad (4.69)$$

The steps of the down sweep phase of the extended inversion algorithm proceed in the same way as in the normal inversion algorithm, but with the following additional operations:

1. After dividing the row associated with the leaf l of the path \mathcal{N}_r by its diagonal element, as in (4.37), we effectuate a row operation similar to (4.39):

$$\begin{aligned} \mathbf{R}(N) &\rightarrow \mathbf{R}(N) - \left(\mathbf{R}(N)|_{\mathcal{N}_r}\right)_L^{\text{LinL}} \mathbf{R}(L) \\ (\bar{\mathbf{B}})_N &\rightarrow (\bar{\mathbf{B}})_N - \left(\mathbf{R}(N)|_{\mathcal{N}_r}\right)_L^{\text{LinL}} (\bar{\mathbf{B}})_L \end{aligned} \quad (4.70)$$

which sets the term $\left(\mathbf{R}(N)|_{\mathcal{N}_r}\right)_L^{\text{LinL}}$ to zero. The remaining non-zero terms in row N are then:

$$\left(\mathbf{R}(N)|_{\mathcal{N}_r}\right)_K^{\text{LinL}} = -\bar{Z}_N I'_{AP} \frac{\alpha_r}{1 - \bar{Z}_L a(\mathcal{N}_r)} \quad (4.71)$$

and

$$\left(\bar{\mathbf{B}}|_{\mathcal{N}_r}\right)_N^{\text{LinL}} = -\bar{Z}_N I'_{AP} \left(\beta_r - \frac{\alpha_r \bar{B}_L}{1 - \bar{Z}_L a(\mathcal{N}_r)} \right). \quad (4.72)$$

2. In step 2, we then define the following additional quantities associated with node P :

$$\alpha_p := \sum_{r \in \mathcal{R}_p} \frac{\alpha_r}{1 - \bar{Z}_L a(\mathcal{N}_r)} \quad (4.73)$$

and

$$\beta_p := \sum_{r \in \mathcal{R}_p} \left(\beta_r - \frac{\alpha_r \bar{B}_L}{1 - \bar{Z}_L a(\mathcal{N}_r)} \right). \quad (4.74)$$

The up sweep phase of the algorithm then proceeds in the same way as before for the nodes

that do not integrate the somatic region.

4.6 Independence and compartmentalization

4.6.1 True independence

The leafs of the NET form the substrate for independence, since in their associated voltage variables there is no direct dependence on the other synaptic conductances. This nevertheless does not mean that they will be independent per se: the SVD at any leaf L still depends on all other nodal voltages on the path to the root:

$$\bar{I}_{g,L}(\bar{\mathbf{V}}) = \sum_{s \in \mathcal{S}_L} g_s f_s(\bar{V}_L + \sum_{N \in \mathcal{N}_L \setminus \{L\}} \bar{V}_N). \quad (4.75)$$

Regarding short term fluctuations, this current will become truly independent from all other synapse if

$$\delta \bar{V}_L > C \sum_{N \in \mathcal{N}_L \setminus \{L\}} \delta \bar{V}_N, \quad (4.76)$$

with C a (large) number that has to be determined empirically and $\delta \bar{V}$ denoting the short term fluctuations of \bar{V} around a long term average $\langle \bar{V} \rangle$. Here, short term means the time-scale on which neurons convert electrical inputs to output. As NMDA synapses have a decay time constant of 20 to 50 ms [Roth and van Rossum, 2009], this time-scale should be at least somewhat larger than the NMDA time constant. We hence take this time-scale to be $\lesssim 100$ ms. Then (4.77) can be approximated as follows:

$$\bar{I}_{g,L}(\bar{V}_L) = \sum_{s \in \mathcal{S}_L} g_s f_s(\bar{V}_L + \langle \sum_{N \in \mathcal{N}_L \setminus \{L\}} \bar{V}_N \rangle). \quad (4.77)$$

The long term average in this equation is only influenced very little by the instantaneous values of the synaptic conductances and can hence be seen as a constant, i.e. a fixed parameter in the equation:

$$\bar{V}_L = \bar{Z}_L(t) * \left[\sum_{s \in \mathcal{S}_L} g_s f_s(\bar{V}_L + \langle \sum_{N \in \mathcal{N}_L \setminus \{L\}} \bar{V}_N \rangle) \right]. \quad (4.78)$$

Consequently, the solution for \bar{V}_L will not depend on the instantaneous values of the synaptic conductances at other locations.

4.6.2 Estimating independence

Whether condition (4.76) holds, depends on the structure of the NET tree as well as the relative size of the synaptic inputs. Since we are chiefly interested in how compartmentalization is

Chapter 4. Dynamic compartmentalization in neurons enables branch-specific learning

induced by the dendritic morphology, we make the rather crude assumption that fluctuations in all synaptic input currents are of a similar size. When this is the case, condition (4.76) becomes a condition on the impedances:

$$\bar{Z}_L > C \sum_{N \in \mathcal{N}_L \setminus \{L\}} \bar{Z}_N |\mathcal{R}_N|, \quad (4.79)$$

where $|\mathcal{R}_N|$ denotes the number of regions node N integrates. When we are interested in determining whether a pair of regions r_i (integrated by the leafs L_i , $i = 1, 2$) can act independently, we can consider a reduced tree with two leafs obtained by pruning all leafs associated with other regions. The new tree then has leaf impedances $\bar{Z}_i = \sum_{N \in \mathcal{N}_{L_i} \setminus (\mathcal{N}_{L_1} \cap \mathcal{N}_{L_2})} \bar{Z}_N$ (i.e. a sum over impedances of nodes that integrate one region but not the other) and a root impedance $\bar{Z}_{12} = \sum_{N \in (\mathcal{N}_{L_1} \cap \mathcal{N}_{L_2})} \bar{Z}_N$ (i.e. a sum over impedances of nodes that integrate both regions). Then, whether one region r_i is independent from the other will be given by

$$\bar{Z}_{L_i} > 2 C \bar{Z}_{12}. \quad (4.80)$$

For mutual independence between r_1 and r_2 , this equation has to hold for both $i = 1$ and $i = 2$. To summarize these two conditions in a single expression, we defined the ‘impedance-inferred independence index’ (I_Z):

$$I_Z = \frac{\bar{Z}_1 + \bar{Z}_2}{2 \bar{Z}_{12}}. \quad (4.81)$$

Then, if (4.80) holds for both regions, the following condition also holds:

$$I_Z > 2 C. \quad (4.82)$$

Note that this is a necessary, but not a sufficient condition for mutual independence. However, as shown throughout the main manuscript, when asymmetry is not too high I_Z is, despite its simplicity, a surprisingly accurate measure. Note that we also checked the effect of asymmetry by computing and ‘asymmetry index’ S_Z :

$$S_Z = \frac{|\bar{Z}_1 - \bar{Z}_2|}{2 \bar{Z}_{12}}. \quad (4.83)$$

In the main text we also checked numerically what the value of C would be: we imposed equation (4.78) on models of varying I_Z and checked the error with respect to the full model. We found that we can speak of independence when $I_Z \gtrsim 10$. This means that $C \approx 5$ in condition (4.76). To conclude, we remark that our discussion on shunting (§ 4.4.4) also holds when multiple excitatory inputs are present on the neuron. Thus, as a shunting conductance will reduce the impedance of the root node, it will also increase I_Z .

4.6.3 Compartmentalization

In § 4.6.1 we discussed the conditions under which a single input site can be considered independent from the rest of the input sites. Nevertheless, when inputs are distributed in an almost continuous fashion along the dendritic arborization, such sites may not exist. It can be expected, however, that the structure of the dendritic tree favors a grouping of inputs, such that inputs belonging to different groups are all mutually independent but inputs belonging to the same group are not. A grouping of this type for homogeneously distributed inputs along the dendritic arborization, and where inputs belonging to different groups have an I_Z above a certain threshold, will be called a compartmentalization of that dendritic tree for that given I_Z . Note that in such a compartmentalization, *not all input sites can belong to a group*, as there will have to be at least some space between compartments if they are to be independent.

How can such a compartmentalization be found? First, we remark that there is no unique answer to this question. Consider a forked dendritic tip. It may happen that inputs within each sister branch are independent from the rest of the dendritic tree, but the branches are not independent from each other. Furthermore, because of a steep impedance gradient within the branch, inputs at the bifurcation point may not be independent from the rest of the tree. Because of the first constraint, both tips can not form separate compartments, whereas because of the second constraint, they can not be grouped into a single compartment either. Hence only one branch can be chosen, and either choice forms a valid compartmentalization.

We implemented an algorithm that proposes, given an I_Z , a compartmentalization that maximizes the number of compartments. Our algorithm uses the NET tree to determine this compartmentalization. We note that if a node N in the NET tree forms a valid compartment, all nodes in the subtree of N are part of the same compartment, since their I_Z to other compartments will be higher than the I_Z of N . Hence, our algorithm will simply return a set of nodes, where it is understood that a compartment associated with a node from this set is its whole subtree. Our algorithm proceeds in three steps:

1. We determine a ‘tentative’ compartmentalization. For each node N in the NET tree, we examine the bifurcation nodes B on the path \mathcal{N}_N from N to the root. We check whether the following condition holds

$$I_Z < \frac{\sum_{K \in \mathcal{N}_N \setminus \mathcal{N}_B} \bar{Z}_K}{\sum_{K \in \mathcal{N}_B} \bar{Z}_K}, \quad (4.84)$$

with \mathcal{N}_b the path from B to the root. If this condition is true for two nodes N and M that have B on their respective paths to the root, and where $(\mathcal{N}_N \setminus \mathcal{N}_B) \cap (\mathcal{N}_M \setminus \mathcal{N}_B) = \emptyset$, these nodes will be separated by at least the required I_Z . Hence, we say that N is a tentative compartment with respect to B .

2. In a second step, we remove all leafs from the tree that could not possibly be separate compartments. To do so, we look at the highest order bifurcation B and its child leafs.

Then, if at least two child leafs are tentative compartments with respect to B , the other leafs who are not tentative compartments with respect to B are removed. Otherwise, all child leafs but the one with largest impedance are removed. Note that in the latter case, B is not a bifurcation anymore and consequently will not induce tentative compartments. We continue to cycle through the bifurcation nodes of highest order until no more nodes can be removed.

3. In a final step we assign the compartments. As we are now sure that every leaf is part of a separate compartment, we start at the leaf, find the nearest bifurcation node in the NET tree, and then recursively find the lowest order node that is still a tentative compartment of B . This node will be a compartment node in the final compartmentalization.

4.7 Simulation-specific parameters

Figure 4.1. We evaluated the impedance matrix in panel B at $4 \mu\text{m}$ intervals. In the simulation depicted in panels D and E, the synapses contained only an AMPA component with $g_{\text{max}} = 10 \text{ nS}$ (representing a cluster of synapses). In the simulations in panels H and K, NMDA synapses were used with $g_{\text{max}} = 2 \text{ nS}$ and $R_{\text{NMDA}} = 3$. In either case, no active channels were inserted in the soma.

Figure 4.2. For the simulations in panel B, NMDA synapses ($R_{\text{NMDA}} = 3$) were used. Their g_{max} and incoming Poisson rate were optimized to utilize the full range of the NMDA non-linearity. For the simulations in the rightmost panels in E-G, 100 ($g_{\text{max}} = 4 \text{ nS}$ and $R_{\text{NMDA}} = 3$) NMDA and 100 GABA ($g_{\text{max}} = 2 \text{ nS}$) synapses were inserted on the morphology and activated with Poisson spike trains of 1 Hz .

Figure 4.3. Traces in panel A corresponded to the simulation parameters in Fig 2B. In panel D, synapse 1 was a non-plastic NMDA synapse ($g_{\text{max}} = 1 \text{ nS}$, $R_{\text{NMDA}} = 3$) and synapse 2 a plastic synapse with the same parameters. When on, synapse 1 received a tonic spike train with a rate of 113 Hz (to yield a time-averaged activation $A_1 \approx 15$). Synapse 2 received rates ranging from 0 to 113 Hz , corresponding to the data points at different activations in the figure in the figure. For the simulations in panel H, we inserted NMDA synapses ($R_{\text{NMDA}} = 3$) in both branches and stimulated them with a rate of 200 Hz . Their maximal conductance g_{max} was optimized to obtain an average depolarization of $-40 \pm 2.5 \text{ mV}$ in each branch.

Figure 4.4. The data analyzed in panels E-G was obtained from the simulations described for Fig 2E-G, ran for 100 s .

Figure 4.5. For the simulations in panel B, the main synapses contained only an AMPA component with $g_{\text{max}} = 5 \text{ nS}$. To simulate the high-conductance state, 200 AMPA and 200

GABA synapses ($g_{\max} = 0.5$ nS) were distributed evenly across the neuron. Each AMPA synapse was stimulated with a Poisson spike train of 5 Hz. The rate of stimulation for the GABA synapses was tuned to achieve a balanced input. To recompute the tree structure for panels A and C, the time-averaged conductances of all background synapses were inserted in the morphology as static shunts.

The inhibitory synapse in panels D-G had $g_{\max} = 0.11$ nS and was activated at a steady rate of 200 Hz, so that its total time-averaged conductance was around 15 nS. For the simulations in panel G, we inserted NMDA synapses ($R_{\text{NMDA}} = 3$) in both branches and stimulated them with a rate of 200 Hz. Note that these inputs could come from multiple presynaptic cells. Due to the linearity of the conductance dynamics however all spikes can be taken to add to the same conductance and can hence be modeled as a single synapse. The maximal conductance g_{\max} of the NMDA synapse was optimized to obtain an average depolarization of -40 ± 2.5 mV in each branch, a target value which yields parameters that allow exploitation of the full range of the NMDA non-linearity.

Figure 4.6. In both simulations with and without shunting inhibition, noise was implemented at all three locations using AMPA ($g_{\max} = 0.1$ nS) and GABA ($g_{\max} = 0.2$ nS) synapses. Both were stimulated with Poisson spike trains of resp. 63 Hz and 157 Hz (tuned to achieve balance). The shunting inhibition in the parent branch was implemented by a GABA synapse ($g_{\max} = 2$ nS) receiving a Poisson train with a rate of 734 Hz (tuned to reach a time-averaged conductance of 31 nS) during the 100 ms learning intervals. Note that this single conductance could again could represent multiple synapses.

Stimulus-specific innervation patterns were: $S(\uparrow)$ to $B(\uparrow)$: 10 synapses, $S(\uparrow)$ to $B(\downarrow)$: 5 synapses, $S(\downarrow)$ to $B(\uparrow)$: 4 synapses and $S(\downarrow)$ to $B(\downarrow)$: 9 synapses. These synapses were all NMDA synapses ($g_{\max} = 1.5$ nS, $R_{\text{NMDA}} = 2$) that were activated in the learning intervals with Poisson trains at a rate of 23 Hz without the shunting inhibition and 33 Hz with the shunting inhibition (to compensate for the loss in input impedance in both branches).

Acknowledgements

WW and MOG were supported by funding from the ETH Domain for the Blue Brain Project (BBP) and the European Union Seventh Framework Program (FP7/2007- 2013) under grant agreements no. FP7-269921 (BrainScaleS), FP7-604102 (THE HUMAN BRAIN PROJECT), as well as EU grant agreement no. 720270 (HBP SGA1). We thank Prof. Wulfram Gerstner and Prof. Idan Segev for helpful advice and discussions and Dr. Luc Guyot and Dr. Aditya Gilra for proofreading the manuscript.

Author contributions

WW and BTN designed the research. WW performed the mathematical modeling, implemented the models and ran the simulations. WW, BTN and MOG analyzed the results and wrote the paper.

5 Conclusion

At the start of this thesis work, the GF formalism was largely abandoned as a tool to simulate and understand dendritic computation. Koch identified three reasons for this [Koch, 1998]: (1) the quadratic scaling with the number of input locations when local synaptic currents depend on the voltage, (2) the cumbersome nature of the convolutions with GF kernels and (3) the apparent inability to integrate non-linear currents. We nevertheless decided to give the GF formalism a second chance, as it contains all the information on how a dendritic tree – no matter how morphologically complex – converts input currents into output voltage. While the GF can indeed only take linear currents into account, it still provides key insights into how non-linear currents interact with the morphology [Segev, 2000]. For instance, Gidon and Segev were able to show that distal inhibition could effectively veto the generation of more proximal dendritic spikes, simply by studying the ratio between transfer and input impedances [Gidon and Segev, 2012].

We started our foray into the Green’s function (GF) formalism by studying whether any advantage could be gained by using it in its most naive form to simulate neuron models. While we found that there was a slight computational advantage with very few synapses on a large morphology [Wybo et al., 2013], the quadratic dependence and cumbersome nature of the convolutions impeded any usefulness for real-world use-cases, where more than a few input sites and possibly a number of non-linear ion channels need to be integrated. We nevertheless noticed that because of the transitivity property of the GF [Koch, 1998], the system of GF integral equations can be rewritten in such a way that most kernels become zero, and the quadratic dependence on the number of input sites becomes linear [Wybo et al., 2015]. To solve the associated convolutions efficiently, we expressed the kernels as sums of exponentials using the vector fitting algorithm [Gustavsen and Semlyen, 1998, Gustavsen and Semlyen, 1999]. These two innovations allowed us to reach a level of computational performance that far exceeds that of compartmental models when only few input locations are present, and can perform at a computational cost that, while slightly higher, is in the same order of magnitude as compartmental models in most realistic use-cases [Wybo et al., 2015]. Nevertheless, our simulation paradigm requires substantially more work to initialize than compartmental models, and

there is no standard software available. These two factors impede its wide-spread adoption. Indirectly though, this work has yielded (and may still yield) very useful results: in [Rössert et al., 2016], the methods we employed in our GF work were used to efficiently extract exponential kernels that take the linear effects of dendrites into account, a key step in the reduction of morphological detailed networks to point neuron networks.

We found however that employing the GF formalism in this way did not improve our understanding of dendritic function, in particular with respect to the subunit hypothesis. For that reason we turned our attention to the separation of variables (SOV) method. In this method, the GF is expressed as an infinite sum of exponentials with a decreasing time-scale, each weighted by a function with a given spatial scale (which also decreases for increasing term numbers) [Major et al., 1993a]. It was our hope that this series could be truncated beyond a specific temporal scale, and that hence the associated spatial functions would provide insight into the spatial extent and location of independent dendritic subunits. This turned out to be unfeasible, as there is a large grey zone between terms in this series that generate the independent subunits and terms that fix the more detailed aspects of the dendritic dynamics. Nevertheless, the SOV method provides a very convenient form to express the GF in, and, as it is a sum of exponentials, convolutions are also reasonably easy to compute.

The first key observation that helped elucidate the subunit hypothesis was that the impedance matrix, obtained by integrating over the temporal coordinate of the GF, is highly structured when dendritic branches are ordered in a depth-first manner [Cuntz et al., 2010]. A second key observation is that this structure is sufficient to derive a tree-like model that captures the full, spatio-temporal dendritic dynamics. We termed this model the neural evaluation tree (NET) and observed that if two regions are independent, this model contains state variables that are truly independent, and thus model the input-output relation of the subunit. We used this observation to derive a condition that determines at what separation on the dendritic tree a pair of input regions can function as independent subunits. The NET also allowed us to quantify how many independent subunits could maximally coexist on the dendritic tree. Furthermore, we noticed that dynamic input regimes, such as the high conductance state, modify the structure of the NET and thus the compartmentalization of the dendrite into subunits. In the high conductance state, this modification is thought to be global, but we also found that targeted shunting inputs can increase compartmentalization locally, in such a way that sister branches that were tightly coupled, can become independent. Branches elsewhere on the dendritic tree do barely feel any difference because of such localized shunting inputs. We finally were able to show that, through this mechanism, branches that would otherwise learn as a single unit, could become responsive to different stimulus features.

These findings are exciting since they could substantially enrich the computational repertoire of neurons. Before, subunits were seen as a static concepts [Häusser and Mel, 2003]. Our results indicate however that the subunits of neurons may be tuned dynamically to a particular computation the network is engaging in. Furthermore, our results provide a possible explanation of recent observations that a large fraction of inhibitory synapses seems

indeed to be targeting intermediate dendritic branches [Bloss et al., 2016], precisely the ones that are optimal to make more distal branches independent. Finally, they also provide insight into the change in input-output relation neurons undergo in-vivo in cortical up-states [Destexhe et al., 2003] and in the high-conductance state [Rudolph and Destexhe, 2003].

Since the NET framework treats synaptic interactions in a comprehensive way, we expect that it will aid in elucidating synaptic interactions that are not yet understood. For instance, the NET could aid in searching the most favorable input patterns along a dendritic tree for eliciting APs in a balanced input regime. That way, it could help elucidating whether it is more beneficial for synaptic inputs of a certain orientation tuning to be spread throughout the dendritic tree [Jia et al., 2010, Chen et al., 2013] or to cluster together [Wilson et al., 2016, Gökçe et al., 2016]. It could also aid in designing an experimental paradigm to directly construct electrical models of neurons. In the past, constructing morphological neuron models was a tedious process, where the full morphology had to be constructed and the parameters of a cable model had to be fitted (see for instance [De Schutter and Bower, 1994a, De Schutter and Bower, 1994b, Hay et al., 2011, Eyal et al., 2016]). Because the NET framework summarizes the electrical structure of a neuron in a drastically simplified form, an approach based on 2-photon glutamate uncaging [Pettit et al., 1997] could be conceived from which a NET model could be derived directly. The NET framework may also help in understanding whether synapses learn primarily through local, voltage-based interactions or rather through backpropagating APs [Sjöström et al., 2008]. Presently, the NET framework only contains point source nonlinearities. While ion channels that are continuously distributed throughout the dendritic tree can be modeled through a re-compartmentalization procedure, not much insight is gained in this way in their localized interactions with synaptic currents. To do so, a comprehensive method that can estimate the ion channel current throughout a dendritic branch following a synaptic input should complement the NET framework.

The NET framework is a promising candidate to enrich neural network models with dendrites in general and dendritic subunits in particular. In its current form, the NET is not particularly efficient to simulate since the kernels at each node are temporally complex. Likely however, the temporal profile of these kernels can be simplified drastically; they could be replaced by a single exponential with only a minor loss in precision. In this form, simulating dendrites would become computationally very efficient. Another benefit of this model is that it would be immediately apparent if the dendrite contained independent subunits or not. Conversely, modelers could devise abstract models, where they choose the degree of independence between synapses as they desire, without having to go through the cumbersome process of choosing the appropriate sites on the dendritic geometry. We aim to turn the tools that have been developed during this thesis into a software toolbox that (1) implements code for the main neural network simulators, such as NEST [Gewaltig and Diesmann, 2007] and BRIAN [Goodman and Brette, 2008, Goodman and Brette, 2009], to simulate the NET efficiently and that (2) provides modelers with the tools to analyse any dendritic tree and extract the NET for any set of input regions. We believe that in light of the rising evidence of the importance of dendritic dynamics in behaving animals [Grienberger et al., 2015, Takahashi et al., 2016,

Chapter 5. Conclusion

Smith et al., 2013, Moore et al., 2017], such a toolbox will be indispensable for modelers aiming to study these phenomena from a computational point of view.

Bibliography

- [Abbott et al., 1991] Abbott, L. F., Farhi, E., and Gutmann, S. (1991). The path integral for dendritic trees. *Biological Cybernetics*, 66(1):49–60.
- [Abrahamsson et al., 2012] Abrahamsson, T., Cathala, L., Matsui, K., Shigemoto, R., and DiGregorio, D. a. (2012). Thin dendrites of cerebellar interneurons confer sublinear synaptic integration and a gradient of short-term plasticity. *Neuron*, 73(6):1159–72.
- [Agmon-Snir et al., 1998] Agmon-Snir, H., Carr, C. E., and Rinzel, J. (1998). The role of dendrites in auditory coincidence detection. *Nature*, 393(6682):268–272.
- [Almog and Korngreen, 2014] Almog, M. and Korngreen, A. (2014). A quantitative description of dendritic conductances and its application to dendritic excitation in layer 5 pyramidal neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 34(1):182–96.
- [Angelo et al., 2007] Angelo, K., London, M., Christensen, S. R., and Häusser, M. (2007). Local and global effects of I(h) distribution in dendrites of mammalian neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(32):8643–53.
- [Antic et al., 2010] Antic, S. D., Zhou, W.-L., Moore, A. R., Short, S. M., and Ikonomu, K. D. (2010). The decade of the dendritic NMDA spike. *Journal of neuroscience research*, 88(14):2991–3001.
- [Archie and Mel, 2000] Archie, K. A. and Mel, B. W. (2000). A model for intradendritic computation of binocular disparity. *Nature neuroscience*, 3(1):54–63.
- [Ascoli, 2006] Ascoli, G. A. (2006). Mobilizing the base of neuroscience data: the case of neuronal morphologies. *Nature reviews. Neuroscience*, 7(4):318–24.
- [Ascoli et al., 2007] Ascoli, G. A., Donohue, D. E., and Halavi, M. (2007). NeuroMorpho.Org: A Central Resource for Neuronal Morphologies. *Journal of Neuroscience*, 27(35):9247–9251.
- [Bayin, 2006] Bayin, S. (2006). *Mathematical Methods in Science and Engineering*. Wiley.
- [Behabadi and Mel, 2014] Behabadi, B. F. and Mel, B. W. (2014). Mechanisms underlying subunit independence in pyramidal neuron dendrites. *Proceedings of the National Academy of Sciences of the United States of America*, 111(1):498–503.

Bibliography

- [Blackman and Tukey, 1958] Blackman, R. and Tukey, J. (1958). *The measurement of power spectra*. Dover publications.
- [Bloss et al., 2016] Bloss, E. B., Cembrowski, M. S., Karsh, B., Colonell, J., Fetter, R. D., and Spruston, N. (2016). Structured Dendritic Inhibition Supports Branch-Selective Integration in CA1 Pyramidal Cells. *Neuron*, 89(5):1016–30.
- [Bono and Clopath, 2016] Bono, J. and Clopath, C. (2016). Beyond spike-timing-dependent plasticity: a computational study of plasticity gradients across basal dendrites . *BioRxiv*, pages 1–26.
- [Branco and Häusser, 2010] Branco, T. and Häusser, M. (2010). The single dendritic branch as a fundamental functional unit in the nervous system. *Current opinion in neurobiology*, 20(4):494–502.
- [Branco and Häusser, 2011] Branco, T. and Häusser, M. (2011). Synaptic integration gradients in single cortical pyramidal cell dendrites. *Neuron*, 69(5):885–92.
- [Bressloff and Coombes, 1997] Bressloff, P. C. and Coombes, S. (1997). Physics of the Extended Neuron. *International Journal of Modern Physics B*, 11(20):2343–2392.
- [Brette et al., 2007] Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., Diesmann, M., Morrison, A., Goodman, P. H., Harris, F. C., Zirpe, M., Natschläger, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A. P., El Boustani, S., and Destexhe, A. (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of computational neuroscience*, 23(3):349–98.
- [Brunel, 2016] Brunel, N. (2016). Is cortical connectivity optimized for storing information? *Nature neuroscience*, 19(5).
- [Bullock and Horridge, 1965] Bullock, T. H. and Horridge, G. A. (1965). *Structure and function in the nervous systems of invertebrates* / [by] Theodore Holmes Bullock and G. Adrian Horridge. With chapters by Howard A. Bern, Irvine R. Hagadorn [and] J. E. Smith. W. H. Freeman San Francisco.
- [Bush and Sejnowski, 1993] Bush, P. C. and Sejnowski, T. J. (1993). Reduced compartmental models of neocortical pyramidal cells. *Journal of Neuroscience Methods*, 46(2):159–166.
- [Butts et al., 2007] Butts, D. a., Weng, C., Jin, J., Yeh, C.-I., Lesica, N. a., Alonso, J.-M., and Stanley, G. B. (2007). Temporal precision in the neural code and the timescales of natural vision. *Nature*, 449(7158):92–96.
- [Butz and Cowan, 1974] Butz, E. G. and Cowan, J. D. (1974). Transient potentials in dendritic systems of arbitrary geometry. *Biophysical journal*, 14:661–689.

- [Carim-Todd et al., 2009] Carim-Todd, L., Bath, K. G., Fulgenzi, G., Yanpallewar, S., Jing, D., Barrick, C. a., Becker, J., Buckley, H., Dorsey, S. G., Lee, F. S., and Tessarollo, L. (2009). Endogenous truncated TrkB.T1 receptor regulates neuronal complexity and TrkB kinase receptor function in vivo. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(3):678–85.
- [Carnevale and Hines, 2006] Carnevale, N. T. and Hines, M. L. (2006). *The NEURON Book*. Cambridge University Press, New York, NY, USA.
- [Caudron et al., 2012] Caudron, Q., Donnelly, S. R., Brand, S. P., and Timofeeva, Y. (2012). Computational convergence of the path integral for real dendritic morphologies. *Journal of mathematical neuroscience*, 2(11):1–28.
- [Chaudhuri and Fiete, 2016] Chaudhuri, R. and Fiete, I. (2016). Computational principles of memory. *Nature Neuroscience*, 19(3):394–403.
- [Chen et al., 2013] Chen, X. R., Heck, N., Lohof, A. M., Rochefort, C., Morel, M.-P., Wehrlé, R., Doulazmi, M., Marty, S., Cannaya, V., Avci, H. X., Mariani, J., Rondi-Reig, L., Vodjdani, G., Sherrard, R. M., Sotelo, C., and Dusart, I. (2013). Mature Purkinje cells require the retinoic acid-related orphan receptor- α (ROR α) to maintain climbing fiber mono-innervation and other adult characteristics. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 33(22):9546–62.
- [Chua et al., 2015] Chua, Y., Morrison, A., and Helias, M. (2015). Modeling the calcium spike as a threshold triggered fixed waveform for synchronous inputs in the fluctuation regime. *Frontiers in computational neuroscience*, 9(July):91.
- [Clopath et al., 2010] Clopath, C., Büsing, L., Vasilaki, E., and Gerstner, W. (2010). Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature neuroscience*, 13(3):344–52.
- [Clopath et al., 2007] Clopath, C., Jolivet, R., Rauch, A., Lüscher, H.-R., and Gerstner, W. (2007). Predicting neuronal activity with simple models of the threshold type: Adaptive Exponential Integrate-and-Fire model with two compartments. *Neurocomputing*, 70(10-12):1668–1673.
- [Coombes et al., 2007] Coombes, A. S., Timofeeva, Y., Svensson, C., Lord, G. J., Josić, K., Cox, S. J., Colbert, C. M., and Josi, K. (2007). Branching dendrites with resonant membrane : A "sum-over-trips" approach. *Biological Cybernetics*, 97(2):137–149.
- [Cuntz et al., 2010] Cuntz, H., Forstner, E., Borst, A., and Häusser, M. (2010). One Rule to Grow Them All: A General Theory of Neuronal Branching and Its Practical Application. *PLoS Computational Biology*, 6(8):e1000877.
- [Davies, 1986] Davies, B. (1986). Locating the zeros of an analytic function. *Journal of computational physics*, 3649(66):36–49.

Bibliography

- [Dayan and Abbott, 2005] Dayan, P. and Abbott, L. F. (2005). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press.
- [De Schutter and Bower, 1994a] De Schutter, E. and Bower, J. M. (1994a). An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice. *Journal of neurophysiology*, 71(1):375–400.
- [De Schutter and Bower, 1994b] De Schutter, E. and Bower, J. M. (1994b). An active membrane model of the cerebellar Purkinje cell II. Simulation of synaptic responses. *Journal of Neurophysiology*, 71(1).
- [Dehghani et al., 2016] Dehghani, N., Peyrache, A., Telenczuk, B., Van Quyen, M. L., Halgren, E., Cash, S. S., Hatsopoulos, N. G., and Destexhe, A. (2016). Dynamic Balance of Excitation and Inhibition in Human and Monkey Neocortex. *Scientific Reports*, 6:23176.
- [Delon and Desolneux, 2007] Delon, J. and Desolneux, A. (2007). A nonparametric approach for histogram segmentation. *IEEE Transactions on Image Processing*, 16(1):253–261.
- [Destexhe et al., 2003] Destexhe, A., Rudolph, M., and Paré, D. (2003). The high-conductance state of neocortical neurons in vivo. *Nature reviews. Neuroscience*, 4(9):739–51.
- [Egger et al., 2015] Egger, R., Schmitt, A. C., Wallace, D. J., Sakmann, B., Oberlaender, M., and Kerr, J. N. D. (2015). Robustness of sensory-evoked excitation is increased by inhibitory inputs to distal apical tuft dendrites. *Proceedings of the National Academy of Sciences of the United States of America*, 112(45):14072–7.
- [Eyal et al., 2016] Eyal, G., Verhoog, M. B., Testa-Silva, G., Deitcher, Y., Lodder, J. C., Benavides-Piccione, R., Morales, J., DeFelipe, J., de Kock, C. P., Mansvelder, H. D., and Segev, I. (2016). Unique membrane properties and enhanced signal processing in human neocortical neurons. *eLife*, 5:e16553.
- [Gerstner et al., 1996] Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding.
- [Gerstner and Kistler, 2002] Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models: An Introduction*. Cambridge University Press, New York, NY, USA.
- [Gerstner et al., 2012] Gerstner, W., Sprekeler, H., and Deco, G. (2012). Theory and Simulation in Neuroscience. *Science*, 338(6103):60–65.
- [Gewaltig and Diesmann, 2007] Gewaltig, M.-O. and Diesmann, M. (2007). Nest (neural simulation tool). *Scholarpedia*, 2(4):1430.
- [Gidon and Segev, 2012] Gidon, A. and Segev, I. (2012). Principles governing the operation of synaptic inhibition in dendrites. *Neuron*, 75(2):330–41.

-
- [Giugliano, 2000] Giugliano, M. (2000). Synthesis of generalized algorithms for the fast computation of synaptic conductances with markov kinetic models in large network simulations. *Neural Computation*, 12(4):903–931.
- [Gökçe et al., 2016] Gökçe, O., Bonhoeffer, T., and Scheuss, V. (2016). Clusters of synaptic inputs on dendrites of layer 5 pyramidal cells in mouse visual cortex. *eLife*, 5:e09222.
- [Golding et al., 2002] Golding, N. L., Staff, N. P., and Spruston, N. (2002). Dendritic spikes as a mechanism for cooperative long-term potentiation. *Nature*, 418(6895):326–331.
- [Goodman and Brette, 2008] Goodman, D. and Brette, R. (2008). Brian: a simulator for spiking neural networks in python. *Frontiers in neuroinformatics*, 2(November):5.
- [Goodman and Brette, 2009] Goodman, D. F. M. and Brette, R. (2009). The brian simulator. *Frontiers in neuroscience*, 3(2):192–7.
- [Govindarajan et al., 2011] Govindarajan, A., Israely, I., Huang, S.-Y., and Tonegawa, S. (2011). The dendritic branch is the preferred integrative unit for protein synthesis-dependent LTP. *Neuron*, 69(1):132–46.
- [Grienberger et al., 2015] Grienberger, C., Chen, X., and Konnerth, A. (2015). Dendritic function in vivo. *Trends in neurosciences*, 38(1):45–54.
- [Gustavsen and Semlyen, 1998] Gustavsen, B. and Semlyen, A. (1998). Simulation of transmission line transients using vector fitting and modal decomposition. *IEEE Transactions Power Delivery*, 13(2):605–614.
- [Gustavsen and Semlyen, 1999] Gustavsen, B. and Semlyen, A. (1999). Rational approximation of frequency domain responses by vector fitting. *IEEE Transaction on Power Delivery*, 14(3):1052–1061.
- [Gütig and Sompolinsky, 2006] Gütig, R. and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. *Nature neuroscience*, 9(3):420–8.
- [Haider et al., 2013] Haider, B., Häusser, M., and Carandini, M. (2013). Inhibition dominates sensory responses in the awake cortex. *Nature*, 493(7430):97–100.
- [Hao et al., 2009] Hao, J., Wang, X.-d., Dan, Y., Poo, M.-m., and Zhang, X.-h. (2009). An arithmetic rule for spatial summation of excitatory and inhibitory inputs in pyramidal neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 106(51):21906–11.
- [Hardie and Spruston, 2009] Hardie, J. and Spruston, N. (2009). Synaptic Depolarization Is More Effective than Back-Propagating Action Potentials during Induction of Associative Long-Term Potentiation in Hippocampal Pyramidal Neurons. *Journal of Neuroscience*, 29(10):3233–3241.

Bibliography

- [Hasselmann, 1988] Hasselmann, K. (1988). PIPs and POPs: The reduction of complex dynamical systems using principal interaction and oscillation patterns. *Journal of Geophysical Research*, 93(D9):11015.
- [Häusser and Mel, 2003] Häusser, M. and Mel, B. (2003). Dendrites: bug or feature? *Current Opinion in Neurobiology*, 13(3):372–383.
- [Häusser et al., 2000] Häusser, M., Spruston, N., Stuart, G. J., and Häusser, M. (2000). Diversity and Dynamics of Dendritic Signalling. *Science*, 290(5492):739–744.
- [Hay et al., 2011] Hay, E., Hill, S., Schürmann, F., Markram, H., and Segev, I. (2011). Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS computational biology*, 7(7):e1002107.
- [Hay et al., 2013] Hay, E., Schurmann, F., Markram, H., and Segev, I. (2013). Preserving axosomatic spiking features despite diverse dendritic morphology. *Journal of neurophysiology*, 109(12):2972–2981.
- [Hendrickx and Dhaene, 2006] Hendrickx, W. and Dhaene, T. (2006). A Discussion of “Rational Approximation of Frequency Domain Responses by Vector Fitting”. *IEEE Transactions on Power Systems*, 21(1):441–443.
- [Hines, 1984] Hines, M. (1984). Efficient computation of branched nerve equations. *International journal of bio-medical computing*, 15(1):69–75.
- [Hines et al., 2004] Hines, M., Morse, T., and Migliore, M. (2004). ModelDB: a database to support computational neuroscience. *Journal of computational neuroscience*, 17(1):7–11.
- [Hodgkin and Huxley, 1952] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544.
- [Holmes et al., 1992] Holmes, W. R., Segev, I., and Rall, W. (1992). Interpretation of time constant and electrotonic length estimates in multicylinder or branched neuronal structures. *Journal of neurophysiology*, 68(4):1401–20.
- [Hong et al., 2012] Hong, S., Ratté, S., Prescott, S. a., and De Schutter, E. (2012). Single neuron firing properties impact correlation-based population coding. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 32(4):1413–28.
- [Horn and Johnson, 2012] Horn, R. A. and Johnson, C. R. (2012). *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 2nd edition.
- [Hubel and Wiesel, 1962] Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106–154.

- [Jack et al., 1975] Jack, J. J., Noble, D., and Tsien, R. W. (1975). *Electric current flow in excitable cells*. Clarendon Press Oxford.
- [Jahr and Stevens, 1990] Jahr, C. E. and Stevens, C. F. (1990). A quantitative description of NMDA receptor-channel kinetic behavior. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 10(6):1830–1837.
- [Jia et al., 2010] Jia, H., Rochefort, N. L., Chen, X., and Konnerth, A. (2010). Dendritic organization of sensory input to cortical neurons in vivo. *Nature*, 464(7293):1307–12.
- [Johanning et al., 2009] Johanning, F. W., Beed, P. S., Trimbuch, T., Bendels, M. H. K., Winterer, J., and Schmitz, D. (2009). Dendritic compartment and neuronal output mode determine pathway-specific long-term potentiation in the piriform cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(43):13649–61.
- [Johnston et al., 1996] Johnston, D., Magee, J. C., Colbert, C. M., and Cristie, B. R. (1996). Active properties of neuronal dendrites. *Annual review of neuroscience*, 19:165–86.
- [Jolivet et al., 2004] Jolivet, R., Lewis, T. J., and Gerstner, W. (2004). Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. *Journal of neurophysiology*, 92(2):959–76.
- [Joyner et al., 1978] Joyner, R. W., Westerfield, M., Moore, J. W., and Stockbridge, N. (1978). A numerical method to model excitable cells. *Biophysical journal*, 22(2):155–70.
- [Kastellakis et al., 2016] Kastellakis, G., Silva, A., and Poirazi, P. (2016). Linking Memories across Time via Neuronal and Dendritic Overlaps in Model Neurons with Active Dendrites. *Cell Reports*, 17(6):1491–1504.
- [Kellems et al., 2010] Kellems, A. R., Chaturantabut, S., Sorensen, D. C., and Cox, S. J. (2010). Morphologically accurate reduced order modeling of spiking neurons. *Journal of computational neuroscience*, 28(3):477–94.
- [Kellems et al., 2009] Kellems, A. R., Roos, D., Xiao, N., and Cox, S. J. (2009). Low-dimensional, morphologically accurate models of subthreshold membrane potential. *Journal of computational neuroscience*, 27(2):161–76.
- [Koch, 1984] Koch, C. (1984). Cable theory in neurons with active, linearized membranes. *Biological Cybernetics*, 50(1):15–33.
- [Koch, 1998] Koch, C. (1998). *Biophysics of Computation: Information Processing in Single Neurons (Computational Neuroscience)*. Oxford University Press, 1 edition.
- [Koch and Poggio, 1985] Koch, C. and Poggio, T. (1985). A simple algorithm for solving the cable equation in dendritic trees of arbitrary geometry. *Journal of neuroscience methods*, 12(4):303–315.

Bibliography

- [Koch et al., 1983] Koch, C., Poggio, T. A., and Torre, V. (1983). Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing. *Proceedings of the National Academy of Sciences of the United States of America*, 80(9):2799–2802.
- [Koch et al., 1982] Koch, C., Poggio, T. A., and Torres, V. (1982). Retinal ganglion cells: a functional interpretation of dendritic morphology. *Philosophical Transactions of the Royal Society of London*, 298(1090):227–263.
- [Kravanja and Van Barel, 2000a] Kravanja, P. and Van Barel, M. (2000a). Clusters of zeros of analytic functions. In *Computing the Zeros of Analytic Functions*, chapter 2, pages 61–81.
- [Kravanja and Van Barel, 2000b] Kravanja, P. and Van Barel, M. (2000b). Zeros and poles of meromorphic functions. In *Computing the Zeros of Analytic Functions (Lecture Notes in Mathematics Volume 1727)*, chapter 3, pages 83–89.
- [Kravanja and Van Barel, 2000c] Kravanja, P. and Van Barel, M. (2000c). Zeros of Analytic Functions. In *Computing the Zeros of Analytic Functions (Lecture Notes in Mathematics Volume 1727)*, chapter 1, pages 1–59.
- [Kuhn et al., 2004] Kuhn, A., Aertsen, A., and Rotter, S. (2004). Neuronal integration of synaptic input in the fluctuation-driven regime. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 24(10):2345–56.
- [Larkum et al., 2007] Larkum, M. E., Waters, J., Sakmann, B., and Helmchen, F. (2007). Dendritic spikes in apical dendrites of neocortical layer 2/3 pyramidal neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(34):8999–9008.
- [Larkum et al., 1999] Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–41.
- [Laudanski et al., 2014] Laudanski, J., Torben-Nielsen, B., Segev, I., and Shamma, S. (2014). Spatially distributed dendritic resonance selectively filters synaptic input. *PLoS computational biology*, 10(8):e1003775.
- [Lerma and De Carlos, 2014] Lerma, J. and De Carlos, J. A. (2014). Epilog: Cajal’s unique and legitimated school. *Frontiers in neuroanatomy*, 8(July):58.
- [London and Häusser, 2005] London, M. and Häusser, M. (2005). Dendritic computation. *Annual review of neuroscience*, 28:503–32.
- [Losonczy et al., 2008] Losonczy, A., Makara, J. K., and Magee, J. C. (2008). Compartmentalized dendritic plasticity and input feature storage in neurons. *Nature*, 452(7186):436–41.
- [Lubich and Schädle, 2002] Lubich, C. and Schädle, A. (2002). Fast convolution for nonreflecting boundary conditions. *SIAM Journal on Scientific Computing*, 24(1):161–182.

- [MacDonald and Wojtowicz, 1982] MacDonald, J. F. and Wojtowicz, J. M. (1982). The effects of l-glutamate and its analogues upon the membrane conductance of central murine neurones in culture. *Canadian Journal of Physiology and Pharmacology*, 60(3):282–296.
- [Magee, 1999] Magee, J. C. (1999). Dendritic Ih normalizes temporal summation in hippocampal CA1 neurons. *Nature neuroscience*, 2(9):508–14.
- [Major, 1993] Major, G. (1993). Solutions for transients in arbitrarily branching cables: III. Voltage clamp problems. *Biophysical journal*, 65(1):469–491.
- [Major and Evans, 1994] Major, G. and Evans, J. D. (1994). Solutions for transients in arbitrarily branching cables: IV. Nonuniform electrical parameters. *Biophysical journal*, 66(3 Pt 1):615–33.
- [Major et al., 1993a] Major, G., Evans, J. D., and Jack, J. B. (1993a). Solutions for transients in arbitrarily branching cables: I. Voltage recording with a somatic shunt. *Biophysical journal*, 65(1):423–49.
- [Major et al., 1993b] Major, G., Evans, J. D., and Jack, J. B. (1993b). Solutions for transients in arbitrarily branching cables II. Voltage clamp theory. *Biophysical journal*, 65(1):469–491.
- [Major et al., 2008] Major, G., Polsky, A., Denk, W., Schiller, J., and Tank, D. W. (2008). Spatiotemporally graded NMDA spike/plateau potentials in basal dendrites of neocortical pyramidal neurons. *Journal of neurophysiology*, 99(5):2584–601.
- [Makara et al., 2009] Makara, J. K., Losonczy, A., Wen, Q., and Magee, J. C. (2009). Experience-dependent compartmentalized dendritic plasticity in rat hippocampal CA1 pyramidal neurons. *Nature neuroscience*, 12(12):1485–1487.
- [Marasco et al., 2012] Marasco, A., Limongiello, A., and Migliore, M. (2012). Fast and accurate low-dimensional reduction of biophysically detailed neuron models. *Scientific reports*, 2:928.
- [Marasco et al., 2013] Marasco, A., Limongiello, A., and Migliore, M. (2013). Using Strahler’s analysis to reduce up to 200-fold the run time of realistic neuron models. *Scientific reports*, 3:2934.
- [Markram, 2006] Markram, H. (2006). The blue brain project. *Nature reviews. Neuroscience*, 7(2):153–60.
- [Markram et al., 2015] Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., Kahou, G. A. A., Berger, T. K., Bilgili, A., Buncic, N., Chalimourda, A., Chindemi, G., Courcol, J.-D., Delalandre, F., Delattre, V., Druckmann, S., Dumusc, R., Dynes, J., Eilemann, S., Gal, E., Gevaert, M. E., Ghobril, J.-P., Gidon, A., Graham, J. W., Gupta, A., Haenel, V., Hay, E., Heinis, T., Hernando, J. B., Hines, M., Kanari, L., Keller, D., Kenyon, J., Khazen, G., Kim, Y., King, J. G., Kisvarday, Z., Kumbhar, P., Lasserre, S., Le Bé, J.-V., Magalhães, B. R. C., Merchán-Pérez,

Bibliography

- A., Meystre, J., Morrice, B. R., Muller, J., Muñoz-Céspedes, A., Muralidhar, S., Muthurasa, K., Nachbaur, D., Newton, T. H., Nolte, M., Ovcharenko, A., Palacios, J., Pastor, L., Perin, R., Ranjan, R., Riachi, I., Rodríguez, J.-R., Riquelme, J. L., Rössert, C., Sfyrakis, K., Shi, Y., Shillcock, J. C., Silberberg, G., Silva, R., Tauheed, F., Telefont, M., Toledo-Rodriguez, M., Tränkler, T., Van Geit, W., Díaz, J. V., Walker, R., Wang, Y., Zaninetta, S. M., DeFelipe, J., Hill, S. L., Segev, I., and Schürmann, F. (2015). Reconstruction and Simulation of Neocortical Microcircuitry. *Cell*, 163(2):456–92.
- [Mathews et al., 2010] Mathews, P. J., Jercog, P. E., Rinzel, J., Scott, L. L., and Golding, N. L. (2010). Control of submillisecond synaptic timing in binaural coincidence detectors by K(v)1 channels. *Nature neuroscience*, 13(5):601–9.
- [Mauro et al., 1970] Mauro, A., Conti, F., Dodge, F., and Schor, R. (1970). Subthreshold behavior and phenomenological impedance of the squid giant axon. *The Journal of general physiology*, 55(4):497–523.
- [Mayer et al., 1984] Mayer, M. L., Westbrook, G. L., and Guthrie, P. B. (1984). Voltage-dependent block by Mg²⁺ of NMDA responses in spinal cord neurones. *Nature*, 309(5965):261–3.
- [Mel, 1993] Mel, B. W. (1993). Synaptic integration in an excitable dendritic tree. *Journal of Neurophysiology*, 70(3):1086–1101.
- [Mel et al., 1998] Mel, B. W., Ruderman, D. L., and Archie, K. A. (1998). Translation-invariant orientation tuning in visual "complex" cells could derive from intradendritic computations. *Journal of Neuroscience*, 18(11):4325–4334.
- [Migliore et al., 1999] Migliore, M., Hoffman, D. a., Magee, J. C., and Johnston, D. (1999). Role of an A-type K⁺ conductance in the back-propagation of action potentials in the dendrites of hippocampal pyramidal neurons. *Journal of computational neuroscience*, 7(1):5–15.
- [Migliore and Shepherd, 2002] Migliore, M. and Shepherd, G. M. (2002). Emerging rules for the distributions of active dendritic conductances. *Nature reviews. Neuroscience*, 3(5):362–70.
- [Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. a., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- [Monahan et al., 2009] Monahan, A. H., Fyfe, J. C., Ambaum, M. H. P., Stephenson, D. B., and North, G. R. (2009). Empirical Orthogonal Functions: The Medium is the Message. *Journal of Climate*, 22(24):6501–6514.
- [Mongillo et al., 2008] Mongillo, G., Barak, O., and Tsodyks, M. (2008). Synaptic theory of working memory. *Science*, 319(5869):1543–1546.

- [Moore et al., 2017] Moore, J. J., Ravassard, P. M., Ho, D., Acharya, L., Kees, A. L., Vuong, C., and Mayank, R. (2017). Dynamics of Cortical Dendritic Membrane Potential and Spikes in Freely Behaving Rats. *Science*, 1497:1–38.
- [Moore et al., 1978] Moore, J. W., Joyner, R. W., Brill, M. H., Waxman, S. D., and Najar-Joa, M. (1978). Simulations of conduction in uniform myelinated fibers. Relative sensitivity to changes in nodal and internodal parameters. *Biophysical journal*, 21(1973):147–160.
- [Müllner et al., 2015] Müllner, F., Wierenga, C., and Bonhoeffer, T. (2015). Precision of Inhibition: Dendritic Inhibition by Individual GABAergic Synapses on Hippocampal Pyramidal Cells Is Confined in Space and Time. *Neuron*, 87(3):576–589.
- [Nemenman et al., 2008] Nemenman, I., Lewen, G. D., Bialek, W., and Van Steveninck, R. R. D. R. (2008). Neural coding of natural stimuli: Information at sub-millisecond resolution. *PLoS Computational Biology*, 4(3):e1000025.
- [Nevian et al., 2007] Nevian, T., Larkum, M. E., Polsky, A., and Schiller, J. (2007). Properties of basal dendrites of layer 5 pyramidal neurons: a direct patch-clamp recording study. *Nature neuroscience*, 10(2):206–14.
- [Norman, 1972] Norman, R. S. (1972). Cable theory for finite length dendritic cylinders with initial and boundary conditions. *Biophysical journal*, 12(1):25–45.
- [Ohme and Schierwagen, 1998] Ohme, M. and Schierwagen, A. (1998). An equivalent cable model for neuronal trees with active membrane. *Biological cybernetics*, 78(3):227–43.
- [Pettit et al., 1997] Pettit, D. L., Wang, S. S. H., Gee, K. R., and Augustine, G. J. (1997). Chemical two-photon uncaging: A novel approach to mapping glutamate receptors. *Neuron*, 19(3):465–471.
- [Pissadaki et al., 2010] Pissadaki, E. K., Sidiropoulou, K., Reczko, M., and Poirazi, P. (2010). Encoding of spatio-temporal input characteristics by a CA1 pyramidal neuron model. *PLoS computational biology*, 6(12):e1001038.
- [Podlaski et al., 2017] Podlaski, W. F., Seeholzer, A., Groschner, L. N., Miesenböck, G., and Vogels, T. P. (2017). ICGenealogy : Mapping the function of neuronal ion channels in model and experiment. *eLife*, 6:e22152:1–22.
- [Poirazi et al., 2003a] Poirazi, P., Brannon, T., and Mel, B. W. (2003a). Arithmetic of subthreshold synaptic summation in a model CA1 pyramidal cell. *Neuron*, 37(6):977–987.
- [Poirazi et al., 2003b] Poirazi, P., Brannon, T., and Mel, B. W. (2003b). Pyramidal neuron as two-layer neural network. *Neuron*, 37(6):989–999.
- [Poirazi and Mel, 2001] Poirazi, P. and Mel, B. W. (2001). Impact of Active Dendrites and Structural Plasticity on the Memory Capacity of Neural Tissue. *Neuron*, 29(3):779–796.

Bibliography

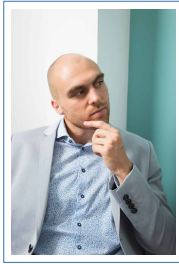
- [Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition.
- [Quarteroni et al., 2016] Quarteroni, A., Manzoni, A., and Negri, F. (2016). *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing.
- [Rabinowitch and Segev, 2006] Rabinowitch, I. and Segev, I. (2006). The interplay between homeostatic synaptic plasticity and functional dendritic compartments. *Journal of neurophysiology*, 96(1):276–83.
- [Rall et al., 1995] Rall, W., Segev, I., Rinzel, J., and Shepherd, G. (1995). *The Theoretical Foundation of Dendritic Function: Selected Papers of Wilfrid Rall with Commentaries*. A Bradford book. Bradford Books.
- [Ratté et al., 2013] Ratté, S., Hong, S., De Schutter, E., and Prescott, S. a. (2013). Impact of neuronal properties on network coding: roles of spike initiation dynamics and robust synchrony transfer. *Neuron*, 78(5):758–72.
- [Rhodes, 2006] Rhodes, P. (2006). The properties and implications of NMDA spikes in neocortical pyramidal cells. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 26(25):6704–15.
- [Richert et al., 2011] Richert, M., Nageswaran, J. M., Dutt, N., and Krichmar, J. L. (2011). An Efficient Simulation Environment for Modeling Large-Scale Cortical Processing. *Frontiers in Neuroinformatics*, 5(September):1–15.
- [Riesenhuber and Poggio, 1999] Riesenhuber, M. and Poggio, T. (1999). Are cortical models really bound by the 'binding problem'? *Neuron*, 24(1):87–93.
- [Rössert et al., 2016] Rössert, C., Pozzorini, C., Chindemi, G., Davison, A. P., Eroev, C., King, J., Newton, T. H., Nolte, M., Ramaswamy, S., Reimann, M. W., Wybo, W. A. M., Gewaltig, M.-O., Gerstner, W., Markram, H., Segev, I., and Muller, E. (2016). Automated point-neuron simplification of data-driven microcircuit models. *arXiv preprint*, pages 1–26.
- [Roth and van Rossum, 2009] Roth, A. and van Rossum, M. C. W. (2009). *Computational Modeling Methods for Neuroscientists*, chapter Modeling Synapses, pages 139–160. MIT Press, Cambridge, MA.
- [Rotter and Diesmann, 1999] Rotter, S. and Diesmann, M. (1999). Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biological cybernetics*, 81(5-6):381–402.
- [Rudolph and Destexhe, 2003] Rudolph, M. and Destexhe, A. (2003). A fast-conducting, stochastic integrative mode for neocortical neurons in vivo. *The Journal of neuroscience*, 23(6):2466–2476.

- [Russell and Norvig, 2003] Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition.
- [Schaefer et al., 2003] Schaefer, A. T., Larkum, M. E., Sakmann, B., Roth, A., Kock, C. P. J. D., Carrel, D., Du, Y., Komlos, D., Hadzimichalis, N. M., Kwon, M., Brzustowicz, L. M., Firestein, B. L., Helmstaedter, M., and Feldmeyer, D. (2003). Coincidence Detection in Pyramidal Neurons Is Tuned by Their Dendritic Branching Pattern. *Journal of Neurophysiology*, 89(February):3143–3154.
- [Schoen et al., 2012] Schoen, A., Salehiomran, A., Larkum, M. E., and Cook, E. P. (2012). A compartmental model of linear resonance and signal transfer in dendrites. *Neural computation*, 24(12):3126–44.
- [Segev, 2000] Segev, I. (2000). Untangling Dendrites with Quantitative Models. *Science*, 290(5492):744–750.
- [Silver, 2010] Silver, A. R. (2010). Neuronal arithmetic. *Nature reviews. Neuroscience*, 11(7):474–89.
- [Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.
- [Sjöström et al., 2008] Sjöström, P. J., Rancz, E. A., Roth, A., and Häusser, M. (2008). Dendritic Excitability and Synaptic Plasticity Dendritic Excitability and Synaptic Plasticity. *Physiological Reviews*, 88(2):769–840.
- [Smith et al., 2013] Smith, S. L., Smith, I. T., Branco, T., and Häusser, M. (2013). Dendritic spikes enhance stimulus selectivity in cortical neurons in vivo. *Nature*, 503(7474):115–20.
- [Spruston, 2008] Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nature reviews. Neuroscience*, 9(3):206–21.
- [Stakgold, 1967] Stakgold, I. (1967). *Boundary value problems of mathematical physics*. Number v. 1 in Macmillan series in advanced mathematics and theoretical physics. Macmillan.
- [Takahashi et al., 2016] Takahashi, N., Oertner, T. G., Hegemann, P., and Larkum, M. E. (2016). Active cortical dendrites modulate perception. *Science*, 354(6319):1587–90.
- [Torben-Nielsen, 2014] Torben-Nielsen, B. (2014). An efficient and extendable python library to analyze neuronal morphologies. *Neuroinformatics*, 12(4):619–22.
- [Torben-Nielsen and Stiefel, 2010] Torben-Nielsen, B. and Stiefel, K. M. (2010). An inverse approach for elucidating dendritic function. *Frontiers in computational neuroscience*, 4(September):128.

Bibliography

- [Traub et al., 2005] Traub, R. D., Contreras, D., Cunningham, M. O., Murray, H., LeBeau, F. E. N., Roopun, A., Bibbig, A., Wilent, W. B., Higley, M. J., and Whittington, M. a. (2005). Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. *Journal of neurophysiology*, 93(4):2194–232.
- [Traub and Miles, 1991] Traub, R. D. and Miles, R. (1991). *Neuronal Networks of the Hippocampus*. Cambridge University Press, New York, NY, USA.
- [Tuckwell, 1988] Tuckwell, H. C. (1988). *Introduction to theoretical neurobiology*. Cambridge studies in mathematical biology, 8. Cambridge University Press.
- [Ulrich, 2002] Ulrich, D. (2002). Dendritic resonance in rat neocortical pyramidal cells. *Journal of neurophysiology*, 87(6):2753–9.
- [Urbanczik and Senn, 2014] Urbanczik, R. and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–8.
- [Van Pelt, 1992] Van Pelt, J. (1992). A simple vector implementation of the Laplace-transformed cable equations in passive dendritic trees. *Biological cybernetics*, 68(1):15–21.
- [Vervaeke et al., 2012] Vervaeke, K., Lorincz, A., Nusser, Z., and Silver, R. A. (2012). Gap junctions compensate for sublinear dendritic integration in an inhibitory network. *Science*, 335(6076):1624–8.
- [Vogels et al., 2011] Vogels, T. P., Sprekeler, H., Zenke, F., Clopath, C., and Gerstner, W. (2011). Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science (New York, N.Y.)*, 334(6062):1569–73.
- [Vreeswijk and Sompolinsky, 1996] Vreeswijk, C. v. and Sompolinsky, H. (1996). Chaos in Neuronal Networks with Balanced Excitatory and Inhibitory Activity. *Science*, 274(5293):1724–1726.
- [Wang et al., 2002] Wang, Y., Gupta, A., Toledo-Rodriguez, M., Wu, C. Z., and Markram, H. (2002). Anatomical, physiological, molecular and circuit properties of nest basket cells in the developing somatosensory cortex. *Cerebral cortex*, 12(4):395–410.
- [Weber et al., 2016] Weber, J. P., Andrásfalvy, B. K., Polito, M., Magó, Á., Ujfalussy, B. B., and Makara, J. K. (2016). Location-dependent synaptic plasticity rules by dendritic spine cooperativity. *Nature Communications*, 7:11380.
- [Wei et al., 2001] Wei, D.-S., Mei, Y.-A., Bagal, A., Kao, J. P. Y., Thompson, S. M., and Tang, C.-M. (2001). Compartmentalized and Binary Behavior of Terminal Dendrites in Hippocampal Pyramidal Neurons. *Science*, 293(5538):2272–2275.
- [Wilson and Kawaguchi, 1996] Wilson, C. J. and Kawaguchi, Y. (1996). The origins of two-state spontaneous membrane potential fluctuations of neostriatal spiny neurons. *Journal of N*, 16(7):2397–410.

- [Wilson et al., 2016] Wilson, D. E., Whitney, D. E., Scholl, B., and Fitzpatrick, D. (2016). Orientation selectivity and the functional clustering of synaptic inputs in primary visual cortex. *Nature Neuroscience*, 19(8):1003–1009.
- [Wu and Mel, 2009] Wu, X. E. and Mel, B. W. (2009). Capacity-enhancing synaptic learning rules in a medial temporal lobe online learning model. *Neuron*, 62(1):31–41.
- [Wybo et al., 2015] Wybo, W. A. M., Boccalini, D., Torben-Nielsen, B., and Gewaltig, M.-O. (2015). A Sparse Reformulation of the Green’s Function Formalism Allows Efficient Simulations of Morphological Neuron Models. *Neural computation*, 27(12):2587–622.
- [Wybo et al., 2013] Wybo, W. A. M., Stiefel, K. M., and Torben-Nielsen, B. (2013). The Green’s function formalism as a bridge between single- and multi-compartmental modeling. *Biological cybernetics*, 107(6):685–694.
- [Zenke et al., 2015] Zenke, F., Agnes, E. J., and Gerstner, W. (2015). Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature communications*, 6:6922.



Willem Wybo

Research Interests

I aim to understand how computational properties of single neurons emerge from the underlying biophysical principles, specifically in dendritic arborizations. I also aim to explore how these properties improve learning and memory in networks of neurons.

Current supervisors Dr. Marc-Oliver Gewaltig and Prof. Henry Markram. In collaboration with Dr. Benjamin Torben-Nielsen.

Education

- 2012–Present **Ph.D. in computational neuroscience**, *EPFL*, Lausanne.
Working on understanding the computational principles of dendritic integration, with a side project on the efficient simulation of the neuron models using the Green's function formalism. Did an internship in the laboratory of Prof. Idan Segev.
- 2010–2012 **M.Sc. in Physics and Astronomy**, *Ghent University*, Ghent, *Greatest Distinction*.
◦ Second year: Took on a strong interest in neuroscience. Completed a master thesis in computational neuroscience and a course project on machine learning that was published as a conference paper.
◦ First year: Exchange year studying theoretical physics at the University of Geneva.
- 2007–2010 **B.Sc. in Physics and Astronomy**, *Ghent University*, Ghent, *Great Distinction*.

Journal publications

- [1] Wybo, W. A. M., Torben-Nielsen, B. and Gewaltig M. O. Dynamic compartmentalization in neurons enables branch-specific learning. *Submitted*.
- [2] Wybo, W. A. M., Boccalini, D., Torben-Nielsen, B. and Gewaltig M. O. A sparse reformulation of the Green's function formalism allows efficient simulations of morphological neuron models. *Neural computation*, 27(12):2587–622, December 2015.
- [3] Wybo, W. A. M., Stiefel, K. M., and Torben-Nielsen, B. The Green's function formalism as a bridge between single- and multi-compartmental modeling. *Biological cybernetics*, 107(6):685–694, September 2013.
- [4] Rössert, C., Pozzorini, C., Chindemi, G., Davison, A. P., Eroo, C., King, J., Newton,

T. H., Nolte, M., Ramaswamy, S., Reimann, M. W., Wybo, W. A. M., Gewaltig, M.-O., Gerstner, W., Markram, H., Segev, I., and Muller, E. Automated point-neuron simplification of data-driven microcircuit models. *arXiv preprint*, arXiv:1604.00087:1–26, August 2016.

Selected conference publications

- [1] Wybo, W. A. M., Boccalini, D., Torben-Nielsen, B. and Gewaltig M. O. A sparse reformulation of the green's function formalism allows efficient simulations of morphological neuron models. *Frontiers in Systems Neuroscience*, (35), 2014.
- [2] Wybo, W. A. M., Torben-Nielsen, B. and Gewaltig M. O. Synaptic synergies and their role in integrating distinct synaptic pathways. *BMC Neuroscience*, 14(Suppl 1):P140, 2013.
- [3] Wybo, W. A. M., Barone R. A., Linaro, D. and Giugliano, M. Modeling the dynamics of excitability and its temperature sensitivity. *Proceedings of the 8th International Meeting on Substrate-Integrated Microelectrode Arrays*, 2012.
- [4] Wybo, W. A. M., Colle, C., Kindermans P.-J. and Schrauwen, B. Distance Dependent Extensions of the Chinese Restaurant Process. *Proceedings of BeneLearn and PMLS 2012*, 2012.

Invited talks

- Title Electrical compartmentalization in neurons
Location ○ Seminar at the Computational Neuroscience Laboratory at Imperial College, London, 2016.

- Title Capturing dendritic computation using the sparse Green's function formalism.
Location(s) ○ HBP CodeJam Workshop #7. Manchester, 2016.
○ Seminar at the biocomputation group at the University of Hertfordshire. Hatfield, 2016.

- Title Application of the separation of variables method to the fly VS network.
Location ○ Max Planck Centre meeting. Munich, 2016.

Master thesis

- Title *Slow Sodium Inactivation and Scale-Invariant Excitability*
- Supervisors Prof. Michele Giugliano (University of Antwerp), Prof. Jan Ryckebusch (Ghent University)
- Description On the modeling of intermittency and scale-free spikerate statistics over extended timescales, as observed by Gal et al., 2010.

Teaching

Assistant for

2013 Machine learning. With Prof. Matthias Seeger.

2013 Analysis. With Prof. Hans-Jörg Ruppen.

2015 Linear Algebra. With Prof. Kathryn Hess.

Supervision

2015 Master thesis Bertrand Marchand: 'Quantifying Spike Prediction Accuracy for Generalized Linear Models.'

Skills

Computational Proficient in analyzing and simulating partial differential equations, model order reduction techniques, matrix analysis, root finding algorithms, various statistical techniques and stochastic processes. A working knowledge of information theory.

Programming Proficient in Python, Matlab and NEURON. A good working knowledge of NEST, C/C++ and linux/unix.

Writing Proficient in scientific writing.

Languages

Dutch Native

English Fluent

French Fluent

References

○ Dr. Marc-Oliver Gewaltig

○ Dr. Benjamin Torben-Nielsen

○ Prof. Wulfram Gerstner

○ Prof. Henry Markram

