

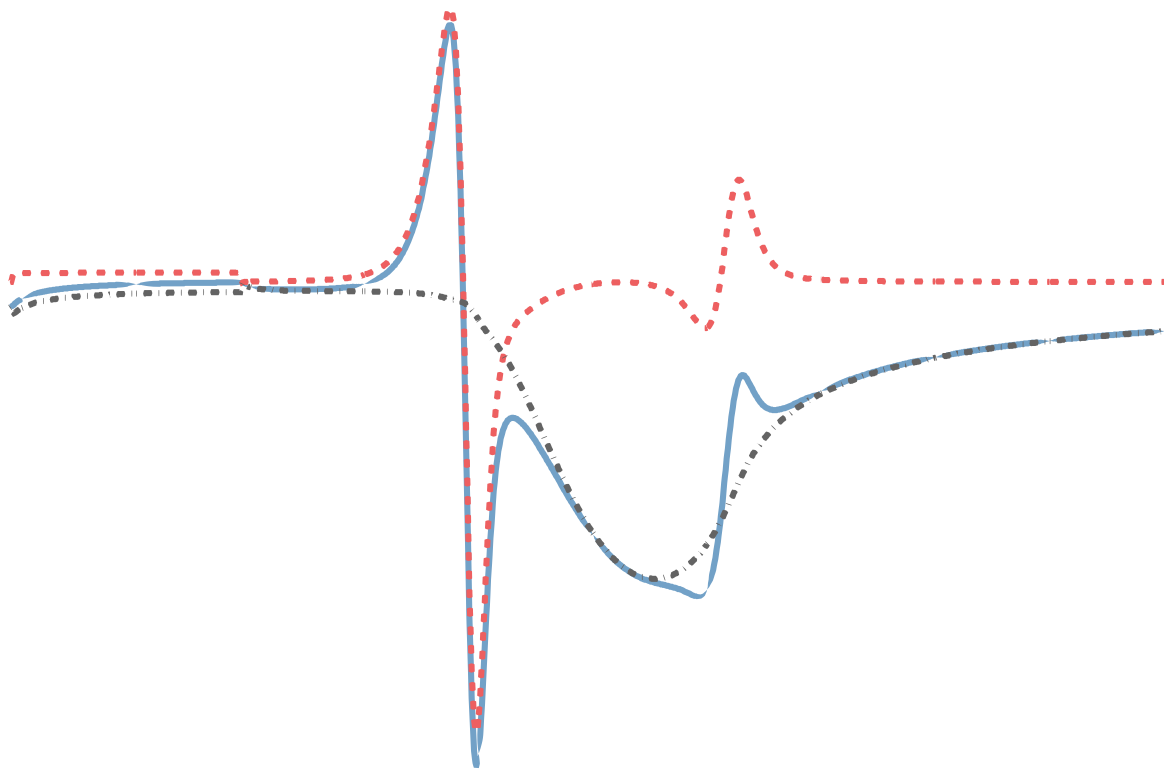
INAUGURAL – DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht – Karls – Universität
Heidelberg

vorgelegt von
Diplom-Informatiker Jurgis Jonas Pods
aus Preetz

Tag der mündlichen Prüfung: 4. Juli 2014

NOTE: This is a slightly modified version with some minor corrections with respect to the dissertation handed in. It was last edited on July 30, 2014. The original version can be found here: <http://www.ub.uni-heidelberg.de/archiv/17128>

Electrodiffusion Models of Axon and Extracellular Space Using the Poisson-Nernst-Planck Equations



Gutachter: Prof. Dr. Peter Bastian

Cover art by [Janneman](#) based on simulation data from this work

Für Paul

«Für deine Verhältnisse nicht schlecht.»

Abstract

In studies of the brain and the nervous system, extracellular signals – as measured by local field potentials (LFPs) or electroencephalography (EEG) – are of capital importance, as they allow to simultaneously obtain data from multiple neurons. The exact biophysical basis of these signals is, however, still not fully understood. Most models for the extracellular potential today are based on volume conductor theory, which assumes that the extracellular fluid is electroneutral and that the only contributions to the electric field are given by membrane currents, which can be imposed as boundary conditions in the mathematical model. This neglects a second, possibly important contributor to the extracellular field: the time- and position-dependent concentrations of ions in the intra- and extracellular fluids.

In this thesis, a 3D model of a single axon in extracellular fluid is presented based on the Poisson-Nernst-Planck (PNP) equations of electrodiffusion. This fundamental model includes not only the potential, but also the concentrations of all participating ion concentrations in a self-consistent way. This enables us to study the propagation of an action potential (AP) along the axonal membrane based on first principles by means of numerical simulations.

By exploiting the cylinder symmetry of this geometry, the problem can be reduced to two dimensions. The numerical solution is implemented in a flexible and efficient way, using the DUNE framework. A suitable mesh generation strategy and a parallelization of the algorithm allow to solve the problem in reasonable time, with a high spatial and temporal resolution. The methods and programming techniques used to deal with the numerical challenges of this multi-scale problem are presented in detail.

Special attention is paid to the Debye layer, the region with strong concentration gradients close to the membrane, which is explicitly resolved by the computational mesh. The focus lies on the evolution of the extracellular electric potential at different membrane distances. Roughly, the extracellular space can be divided into three distinct regions: first, the distant farfield, which exhibits a characteristic triphasic waveform in response to an action potential traveling along the membrane. This is consistent with previous modeling efforts and experiments. Secondly, the Debye layer close to the membrane, which shows a completely different extracellular response in the form of an “AP echo”, which is also observed in juxtacellular recordings. Finally, there is the intermediate or diffusion layer located in between, which shows a gradual transition from the Debye layer potential towards the farfield potential. Both of these potential regions show marked deviations from volume conductor models, which can be

attributed to the redistribution of concentrations and associated ion fluxes. These differences are explained by analyzing the capacitive and ionic components of the potential.

In an extension, we also include myelination into the model, which has a significant impact on the extracellular field. Again, the numerical results are compared to volume conductor models.

Finally, a model study is carried out to assess the magnitude of ephaptic effects, i.e. the influence of the electric field of one cell on a neighboring cell, in a somewhat artificial geometry. While the results probably can not be interpreted quantitatively in the majority of physiological situations, the qualitative behavior shows interesting effects. An axon can elicit an action potential in a surrounding bundle of axons, given that the distance is small enough and the resistivity of the extracellular medium is significantly increased. Further results of this study are extremely large extracellular potentials with amplitudes up to 100 mV and an unusual neuronal firing mode in which the cell is not depolarized by an increase in the intracellular potential, but by a decrease in the extracellular potential. Some literature references are given that show that these observations are consistent with previous studies.

Zusammenfassung

In Studien des Gehirns und des Nervensystems sind extrazelluläre Signale – gemessen in der Form von Local Field Potentials (LFPs) oder als Elektroenzephalografie (EEG) – von großer Bedeutung, da sie die simultane Erhebung von Daten mehrerer Neuronen erlauben. Die genaue biophysikalische Grundlage dieser Signale ist jedoch noch immer nicht vollständig verstanden. Heutzutage basieren die meisten Modelle für das extrazelluläre Potential auf dem Konzept von “Volume Conductors” (Volumenleitern), bei denen angenommen wird, dass die extrazelluläre Flüssigkeit elektroneutral ist und dass die einzigen Beiträge zum extrazellulären Feld von Membranströmen stammen, die als Randbedingungen ins mathematische Modell Einzug finden. Dies vernachlässigt einen zweiten, möglicherweise bedeutenden Beitrag zum extrazellulären Feld: Die zeit- und ortsabhängigen Konzentrationen der in den intra- und extrazellulären Fluiden befindlichen Ionen.

In dieser Arbeit wird ein 3D-Modell eines Axons in extrazellulärer Flüssigkeit auf Basis der Poisson-Nernst-Planck Gleichungen der Elektrodifffusion präsentiert. Dieses fundamentale Modell beinhaltet das Potential und zudem die Konzentrationen aller beteiligten Ionen in einer selbstkonsistenten Art und Weise. Dies ermöglicht es, die Ausbreitung eines Aktionspotentials (AP) entlang der axonalen Membran auf Basis grundlegender physikalischer Gesetze mit den Mitteln der numerischen Simulation zu studieren.

Durch das Ausnutzen der Zylindersymmetrie der vorliegenden Geometrie kann das Problem auf zwei Dimensionen reduziert werden. Die numerische Lösung ist in einer flexiblen und effizienten Weise unter Verwendung des DUNE-Frameworks implementiert. Eine geeignete Gittergenerierungsstrategie und eine Parallelisierung des numerischen Algorithmus erlauben es, das Problem in angemessener Zeit und mit einer hohen räumlichen und zeitlichen Auflösung zu lösen. Die verwendeten Methoden und Programmier Techniken zur Überwindung der numerischen Herausforderungen dieses Multiskalen-Problems sind detailliert dargestellt.

Besondere Beachtung gilt dabei dem Debye-Layer, der Region nahe der Membran, innerhalb der die Konzentrationen starke Gradienten aufweisen. Diese Region ist im Rechengitter explizit aufgelöst. Das Hauptaugenmerk liegt auf der Entwicklung des extrazellulären elektrischen Potentials bei unterschiedlichen Abständen von der Membran. Der Extrazellulärraum kann grob in drei verschiedene Bereiche aufgeteilt werden: Erstens das Fernfeld, in welchem die Wellenform einen charakteristischen dreiphasigen Verlauf aufweist. Dies ist konsistent mit vorhergehenden Modellierungsversuchen. Zweitens der Debye-Layer nahe der Membran, welcher eine gänzlich andere extrazelluläre Antwort in Form eines “AP-Echos”

aufweist, welches auch in juxtazellulären Messungen beobachtet wird. Zwischen diesen beiden Regionen befindet sich die intermediäre oder Diffusions-Schicht, welche einen graduellen Übergang vom Potential des Debye-Layers zu dem des Fernfeldes zeigt. Diese beiden letzten Regionen zeigen deutliche Abweichungen von Volume Conductor-Modellen, die auf Konzentrationsumverteilungen und die damit einhergehenden Ionenflüsse zurückgeführt werden können. Diese Unterschiede werden durch Analyse der einzelnen, kapazitiven und ionischen Beiträge zum Potential erklärt.

In einer Erweiterung wird auch eine Myelinisierung und deren signifikanter Einfluss auf das extrazelluläre Feld im Modell berücksichtigt. Die Resultate werden wiederum mit Volume-Conductor-Modellen verglichen.

Schließlich wird eine weitere Modellstudie durchgeführt, die der Beurteilung von ephaptischen Effekten dient, also der Beeinflussbarkeit einer Zelle durch das elektrische Feld einer benachbarten Zelle, was unter Benutzung einer recht artifiziellen Geometrie simuliert wird. Obwohl die Ergebnisse für die Mehrzahl der physiologischen Situationen wahrscheinlich nicht quantitativ interpretiert werden können, zeigen sie dennoch interessante qualitative Effekte. Ein Axon kann ein Aktionspotential in einem umgebenden Axonbündel auslösen, solange die Distanz zwischen beiden klein genug und die Resistivität des Extrazellulärraums stark erhöht ist. Weitere Ergebnisse dieser Studie sind extrem große extrazelluläre Potentiale mit Amplituden bis zu 100 mV und ein ungewöhnlicher Feuermodus, bei dem eine Zelle nicht durch die Zunahme des intrazellulären Potentials, sondern durch die Abnahme des extrazellulären Potentials depolarisiert wird. Dazu werden einige Referenzen aus der Literatur angegeben, die zeigen, dass diese Beobachtungen konsistent mit vorausgehenden Studien sind.

Contents

List of Figures	V
List of Tables	VIII
1 Introduction	1
1.1 Motivation	1
1.2 Challenges for Scientific Computing in Computational Neuroscience	4
1.3 Related Work	6
1.4 Outline	9
2 Theory	11
2.1 Mathematical Aspects of Partial Differential Equations	11
2.1.1 Type Classification	11
2.1.2 Prototypes	12
2.1.2.1 The Poisson Equation	12
2.1.2.2 The Heat Equation	13
2.1.2.3 The Transport Equation	13
2.1.2.4 The Convection-Diffusion Equation	14
2.2 Numerical Solution of PDEs by the Finite Element Method	14
2.2.1 Triangulation of the Domain	15
2.2.2 The Finite Element Method	15
2.2.2.1 Weak Form	15
2.2.2.2 Existence and Uniqueness of Weak Solutions	17
2.2.2.3 The Continuous Galerkin Method	18
2.2.2.4 Galerkin Orthogonality	20
2.2.2.5 Common Function Spaces	21
2.2.3 Nonlinear PDEs: Newton's Method	22
2.2.4 Time Discretization	23
2.2.5 Linear Solvers	24
2.3 Fundamentals of Biophysics and Neuroscience	24
2.3.1 Structure and Function of Neurons	24

2.3.2	Membrane Physics	26
2.3.2.1	Poisson-Boltzmann Equation	27
2.3.2.2	Debye-Hückel Theory	28
2.3.3	Ion Channels	29
2.3.3.1	Leak Channels	29
2.3.3.2	The Hodgkin-Huxley System of Membrane Excitation	32
2.3.4	Cable Equation	35
2.3.5	Extracellular Space	36
2.3.5.1	Volume Conductor Theory	36
2.3.5.2	Line Source Approximation	37
2.3.5.3	The Poisson-Nernst-Planck Equations of Electrodiffusion	38
3	General Model of an Axon in Extracellular Space	41
3.1	General Remarks	41
3.2	A Simplified Axon Model in Cylindrical Coordinates	42
3.3	PNP Equations and Boundary Conditions	42
3.4	Derivation of the Membrane Flux	46
3.5	Numerical Methods	48
3.5.1	Weak Form and Discretization	48
3.5.1.1	Space Discretization	49
3.5.1.2	Time Discretization	50
3.5.2	Solving the PDE System	51
3.5.2.1	Linear Solvers	52
3.5.3	Validation of the Numerical Algorithm	52
4	Implementation	58
4.1	The DUNE Framework	58
4.2	The DUNE Module <code>dune-ax1</code>	60
4.2.1	Directory Structure	60
4.2.2	Main Components	61
	<code>acme2_cyl_par.cc</code>	61
	<code>Ax1GridGenerator</code>	61
	<code>Acme2CylFactory</code>	62
	<code>Acme2CylPhysics</code>	63
	<code>Acme2CylGeometrySwitch</code>	63
	<code>Acme2CylGeometry</code>	64
	<code>Acme2CylSetup</code>	65

	Acme2CylSimulation	66
	Acme2CylOutput	68
5	Model of an Unmyelinated Axon	69
5.1	Simulation Setup	69
5.1.1	Parallelization	71
5.1.2	Linear Solver and Numerical Performance	72
5.2	Simulation Results	74
5.2.1	Equilibrium States	74
5.2.2	Action Potential	75
5.2.2.1	Intracellular Potential	77
5.2.2.2	Membrane Flux	78
5.2.2.3	Near- and Farfield Extracellular Potential	80
5.2.2.4	Debye Layer Extracellular Potential	84
5.2.2.5	Transition Interval from Debye Layer to Nearfield	87
5.2.2.6	Extracellular Concentrations	91
5.3	Comparison with Other Models	93
5.3.1	Line Source Approximation	93
5.3.1.1	Separate Inspection of EAP Contributions	94
5.3.2	Numerical Solution of the Volume Conductor Equation	98
5.4	Summary	101
6	Model of a Myelinated Axon	102
6.1	Representation of Myelin in the Model	102
6.2	Simulation Setup	106
6.2.1	Modified Grid Generation	106
6.2.2	Permittivity Smoothing	108
6.2.3	Parallelization and Adjustments of Numerical Methods	108
6.2.4	Solver Scaling Tests	110
6.3	Simulation Results	112
6.3.1	Some Notes on the Separate Visualization of Myelin and Node Values	113
6.3.2	Equilibrium States	113
6.3.3	Action Potential	115
6.3.3.1	Intracellular Potential	115
6.3.3.2	Membrane Flux	115
6.3.3.3	Debye Layer Concentrations	117
6.3.3.4	Debye Layer Extracellular Potential	118
6.3.3.5	Near- and Farfield Extracellular Potential	119

6.4	Comparison with LSA	120
6.5	Summary	123
7	Ephaptic Interaction Between Multiple Axon Fibers	126
7.1	Previous Work	126
7.2	Unmyelinated Axon Surrounded by an Axon Bundle	127
7.2.1	Geometry	127
7.2.2	Action Potential	130
7.2.2.1	Fixed bundle diameter d_b	132
7.2.2.2	Volume-corrected bundle diameter d_b	138
7.3	Summary	140
8	Summary and Discussion	141
	Bibliography	148
A	Results of the Volume-Corrected Axon Bundle Simulations	159
	Acronyms	164

List of Figures

2.1	Illustration of a typical neuron	25
2.2	Neuronal signaling mechanisms	26
2.3	Schematic picture of a membrane and electrolyte	27
2.4	Equivalent circuit for a patch of membrane	30
2.5	Equivalent circuit for the Hodgkin-Huxley membrane patch model	32
2.6	Equivalent circuit for a spatially extended neuronal membrane	35
3.1	Two-dimensional computational domain for the cylinder-symmetric axon model	43
3.2	Solution for hamburger example at different time points	53
3.3	Hamburger convergence behavior in space	54
3.4	Hamburger convergence behavior in time	54
3.5	Solution for cheeseburger example at different time points	55
3.6	Cheeseburger convergence behavior for spatial discretization	55
3.7	Cheeseburger convergence behavior in time	56
3.8	Solution for bigmac example at different time points	56
3.9	Bigmac convergence behavior for spatial discretization	57
3.10	Bigmac convergence behavior in time	57
4.1	Comparison of matrix structure depending on DOF ordering	67
5.1	Partition of the unmyelinated axon computational domain for the parallel case	72
5.2	Equilibrium charge density	76
5.3	Equilibration of membrane fluxes	76
5.4	Action potentials evoked at equidistant positions along the axon	78
5.5	Different components of the membrane flux at fixed point on the membrane	80
5.6	Ionic and capacitive membrane flux at fixed point on the membrane	81
5.7	Total membrane flux at fixed point on the membrane	81
5.8	Extracellular potential at various distances from the membrane	82
5.9	Snapshot of the nearfield extracellular potential profile	83

5.10	Snapshot of the whole extracellular potential	84
5.11	Snapshot of the Debye layer extracellular potential profile	86
5.12	Extracellular potential time courses over Debye layer	88
5.13	Potential profile over Debye layer at various time points	89
5.14	Decay of peak-to-baseline in EAP over Debye layer	90
5.15	Charge density profile over Debye layer at various time points	90
5.16	Snapshot of the Debye layer and nearfield concentration profiles	92
5.17	Comparison of ED and LSA potentials	95
5.18	Comparison of the capacitive component of the EAP for ED and LSA	96
5.19	Comparison of the ionic component of the EAP for ED and LSA	97
5.20	Comparison of numerical and LSA solution to the volume conductor equation	99
5.21	Comparison of ED and VC results	100
6.1	Computational domain with explicit myelin representation	103
6.2	Comparison of potential profiles for different membrane resolutions	104
6.3	Relation of linear potential decay and membrane thickness/permittivity	105
6.4	Computational domain with collapsed myelin representation	105
6.5	Equilibrium permittivity profile	109
6.6	Partition of the computational domain for the parallel case	110
6.7	Initial defect for different domain sizes	111
6.8	Equilibrium charge density profile	114
6.9	Intracellular action potential propagation	116
6.10	Membrane flux at node of Ranvier and myelin	116
6.11	Debye layer concentration time courses at node of Ranvier	117
6.12	Debye layer potential time course at node of Ranvier	118
6.13	Debye layer potential time course at myelin	119
6.14	Extracellular potential time courses at various positions	121
6.15	Comparison with LSA at various farfield positions	122
6.16	Comparison with LSA: zoom into myelin-node transition and nearfield	124
7.1	Computational domain for multiple axons	128
7.2	Illustration of axonal packing in different geometric arrangements	130
7.3	Ratios of intra- and extracellular volumes for different axon-bundle distances	131
7.4	Membrane potentials for different axon distances d and fixed bundle diameter d_b	133

7.5	Potential and charge density time courses for $d = 1 \mu\text{m}$	135
7.6	Potential and charge density time courses for $d = 500 \text{ nm}$	136
7.7	Potential and charge density time courses for $d = 100 \text{ nm}$	137
7.8	Potential and charge density time courses for $d = 50 \text{ nm}$	138
7.9	Membrane potentials for different axon distances d and adjusted bundle diameters d_b (volume-corrected)	139
A.1	Potential and charge density time courses for $d = 1 \mu\text{m}$ (volume-corrected)	160
A.2	Potential and charge density time courses for $d = 500 \text{ nm}$ (volume-corrected)	161
A.3	Potential and charge density time courses for $d = 100 \text{ nm}$ (volume-corrected)	162
A.4	Potential and charge density time courses for $d = 50 \text{ nm}$ (volume-corrected)	163

List of Tables

5.1	Simulation parameters for the unmyelinated axon model	70
5.2	Simulation timings for the parallel solution using different processor counts p	73
5.3	Solver statistics per time step for the parallel solution using different processor counts p	73
5.4	Relative leak conductances and resulting equilibrium membrane potentials	75
6.1	Simulation parameters for the myelinated axon model	107
6.2	Solver statistics per time step for the parallel solution using different processor counts p	112
7.1	Simulation timings for the axon bundle setup using different inter-fiber distances d and a fixed bundle diameter d_b	132
A.1	Simulation timings for the axon bundle setup using different inter-fiber distances d and a volume-corrected bundle diameter d_b	159

1 Introduction

If we knew what it was we were doing, it would not be called research, would it?

(*Albert Einstein*)

1.1 Motivation

When opening google.com and typing in the word “computational”, the first two occurrences the auto-complete feature gives you are “computational fluid dynamics” and “computational biology” (as of December 19, 2013). While this might in no way be representative, it hints at the popularity the application of computational methods in biology has gained. Recently, the field of *computational neuroscience* has been established, which strives to use methods from computational sciences for the understanding of the brain and the nervous system.

Why is this? Biology is a particular example of a field that has always been dominated by experimental methods. This is not astonishing, since it deals with the study of living organisms, which are complex and are hard to describe in a theoretical framework. Observations have therefore been the primary source of knowledge until today. However, for certain biological disciplines like systems biology or molecular biology, mathematical models have been quite successful in explaining and predicting the behavior of the underlying system, thereby complementing experiments as a second pillar.

When it comes to complicated systems, the corresponding mathematical models can not be evaluated by hand, either because the model – mostly in the form of differential equations – is too complex, or simply because there is no known analytical solution. For these cases, the computer has proven to be a powerful tool for solving even the most difficult problems.

Computers can be used to carry out complicated calculations within a fraction of a second, which in former times would have taken great mathematicians years. Consequently, the computer has become the single most important tool in the majority of mathematical divisions. Computer simulations based on the rigorous mathematical foundation of numerical analysis have established as a third pillar in addition to theory and experiment.

In biology, the importance of numerical simulations is even reflected in the terminology, where the phrases *in vivo* (in the living organism) and *in vitro* (in the test tube) for the description of an experiment have been extended by *in silico* (in the computer), showing that simulations are regarded as another form of experiment.

Neuroscience, as a special case, imposes additional difficulties on experiments, as the nervous system and particularly the human brain are especially complex systems. Neurons, the main building blocks, show complicated morphologies, extending and intertwining and eventually forming connections between each other – the synapses. It is estimated that the average human brain consists of about 85 billion neurons, about 20 billion of which are located in the neocortex, the outer layers of the brain. On average, each of these neurons forms 7000 connections to other neurons, resulting in the impressive number of 1.5 trillion synapses for the neocortex alone, in a volume of about 1.5l [35]. This makes the brain a very inaccessible organ rendering experimental investigations extremely difficult.

It is also notable that the availability of intact human brain tissue naturally is very restricted, so most of the experimental studies have been carried out in other mammals such as mice and rats, whose brains show similarities to the human one.

Considering all this, it is clear why *in silico* investigations in the form of computer simulations have become increasingly popular. But there is more to it: the brain computes, as does the computer – although in a very different way. Both share the computation on an abstract level, i.e. transforming a certain set of inputs into a set of outputs. While the circuit between input and output may look completely different for a set of neurons and a desktop computer, both have the fundamental property of implementing a transfer function from input to output data, with an underlying *encoding* of the data. While this might seem trivial on first sight, it has shown to be the hardest question of all: how is information encoded in the brain?

For a computer, we know the numeral system is binary, we know how to map sequences of bits to data types, how to (approximately) represent real numbers as a bit string, and how to tell the processor to direct data through transistor gates to implement a certain function. This is everything but trivial, but the process is completely transparent, as human beings invented the concepts and built them up from scratch, documenting structure and function down to the lowest level. The brain, in contrast, was built by nature, so its structure and function have to be reverse-engineered.

In fact, experiments can tell us something about the transfer function between

input and output, but, to stay in the picture of a digital computer, telling that an input bit sequence of 1010001010 is being transformed to a sequence of 100101 does not provide us with any knowledge, as long as we do not know what those sequences *mean*. The application of information theory and data analysis on experimental data, although still in its infancy, has already proven extremely useful, and it can be expected that these methods will strongly shape the future of neuroscience.

At present, the field of neuroscience is booming, reflected in billion-dollar initiatives like the European Human Brain Project [71] or its American counterpart BRAIN [58]. Still, the code of the brain has not been deciphered until now. Joint efforts involving experimental and theoretical work will be required to systematically uncover structure and function of the brain, starting at the elementary unit, the single neuron.

In this thesis we will focus on the *axon*, the part of the neuron responsible for generating the fundamental data unit of the brain, the [action potential \(AP\)](#). We can think of it as a bit for now, but – as we will see later – it is not exactly binary, as it can be shaped to a large degree by several physiological mechanisms. In this work we are, however, not primarily interested in the action potential itself, but rather in the electric field it generates extracellularly. The extracellular potential is of special interest, as many experiments today rely on extracellular recordings which allow to capture signals called [local field potentials \(LFPs\)](#) of a moderately large amount of nearby cells without the need to place the electrode into the soma of a neuron, as it is done in intracellular recordings. Extracellular recordings are very useful for the investigation of the interplay of a network of neurons, particularly “brain waves” emerging from synchronized firing of a large number of neurons. These network oscillations can be observed *in vivo* as well as *in vitro*, but not only by (invasive) electrode measurements, but also by noninvasive [electroencephalography \(EEG\)](#). They are most commonly classified according to their frequency, e.g. theta (4 Hz–7 Hz), alpha (8 Hz–13 Hz), beta (12 Hz–30 Hz), and gamma (25 Hz–100 Hz).

Since these rhythms are used very frequently to explain network behavior of the brain – and, as such, are also involved in deciphering the code of the brain – it is crucial to understand the origin of these signals. This thesis strives to elucidate how extracellular signals are generated based on first principles by setting up a detailed model involving the movement of ions, the elementary charge carriers responsible for any electric activity in living organisms.

1.2 Challenges for Scientific Computing in Computational Neuroscience

Above, we introduced the aspiring field of computational neuroscience. The name implies the application of computational methods to the field of neuroscience. But it does not pinpoint the actual character of the methods involved. In fact, many projects in neuroscience are being relabeled and equipped with the trendy tag “computational”, thus acknowledging that the used software for carrying out statistical evaluations indeed involves computational methods. However, the use of truly new computational methods in the sense of scientific computing involves conceptual mathematic modeling and computer simulations, which – depending on the complexity of these models – requires special numerical methods. It is clear that the numerical simulation of complex biological models is not covered by the domain of biology anymore, since it requires deeper knowledge in e.g. mathematics, physics, and computer science.

The number of publications that use computer simulations as a complement to experiments is steeply increasing. Still, many of these models are simplistic and based on the availability of simulation tools such as NEURON [53]. Sometimes it seems the models were merely fitted to reproduce the experimental data, and not for their ability to explore situations which are not feasible in experimental setups, or to make predictions which can be validated or falsified in experiments. Likely, this is caused by the lack of expertise in scientific computing, therefore limiting the complexity of models and the accuracy of simulations.

There are surprisingly few projects that actually employ [high performance computing \(HPC\)](#), the discipline that deals with parallelized simulations running on supercomputers. The underlying mathematical models often have systems with millions or even billions of [degrees of freedom \(DOFs\)](#), such that they are not solvable sequentially in reasonable time. This is the boundary that many neuroscientific investigations do not dare to cross, as the parallelization of numerical codes is cumbersome and requires special knowledge in both numerics and programming. One of the few [HPC](#) projects in neuroscience is the Blue Brain Project [70], which is now part of the EU initiative Human Brain Project. IBM sponsored one of their Blue Gene supercomputers for the purpose of building accurate models of the human brain based on a parallelized version of the NEURON simulator. This project can therefore be seen as a lighthouse project which dares to approach the challenge of tackling simulations with over a hundred million connected neurons. The list of comparable projects using current methods from scientific computing is small.

The low visibility of scientific computing in neuroscience – as opposed to many disciplines in physics, chemistry or engineering, like molecular dynamics, groundwater flow or aerodynamics – should be seen as an encouragement. It is not that methods of scientific computing are not needed in biology and neuroscience, the opposite is the case: the nervous system is elaborate and delicate, with many levels of abstraction from the macroscopic level of brain regions down to the subcellular molecular level. On each of these levels, models of different complexity may be employed. Time-dependent nonlinear [partial differential equations \(PDEs\)](#) in three dimensions usually require parallelization in order to achieve reasonable accuracy and resolution of the given domain, as the number of [DOFs](#) quickly reaches the order of one million or more, where a sequential solution would take too long. This involves techniques such as domain decomposition, suitable parallel linear solvers and preconditioners, and parallel [input/output \(IO\)](#) for writing out the results. A lot of experience and knowledge has been acquired in scientific computing in other application domains, and it seems beneficial to transfer the applied methods to the field of neuroscience.

In the present work, we will deal with a nonlinearly coupled system of [PDEs](#) in a three-dimensional setting. Even after the introduction of a cylinder geometry and the resulting complexity reduction to a 2D problem, large numbers of unknowns ranging roughly between 10^5 and 10^7 are required. This necessitates an efficient parallelization scheme to achieve acceptable simulation times. We will see that not only the large number of [DOFs](#), but also the inherent physical properties of the [PDE](#) system provide substantial difficulties for the numerical solution of the model. One central point will be the electrical property of the membrane acting as a capacitance, which necessitates a fine spatial resolution in perpendicular direction, while the direction parallel to the membrane may be resolved to a much coarser degree. This results in a trade-off between the effort to reduce the number of unknowns and the emerging anisotropy in the spatial resolution. It is known that an anisotropic grid imposes severe difficulties on the numerical solution, as the linear system to be solved involves dealing with a badly conditioned matrix. This holds true even more in a parallel computation, since interior domain borders (marking processor boundaries) need to be treated with special care.

For all these reasons, the development of a proper mathematical model together with an accurate and efficient numerical solution is a topic on its own and very much worth investigating in the context of scientific computing. Hopefully, the insights gained throughout this thesis will not only contribute to the understanding of neuronal signal propagation in the neuroscientific context, but also to the field of scientific and high performance computing, as the methods used might also be applied to related models with similar physical and numerical properties.

Jointly, and most importantly from a personal point of view, we hope that this thesis might serve as a small step and future motivation to bring the fields of neuroscience and scientific computing closer together, as in the future many open problems in neuroscience might be solved by the application of scientific computing and its powerful computational methods.

1.3 Related Work

Simulations of neuronal signal propagation have a long history and are well established in neuroscience as a useful tool for the study of brain function. Most models today are based on the seminal work of Hodgkin and Huxley [55] for the dynamics of membrane currents and the application of *cable theory* [84] to account for the tree-like neuron morphology. In essence, the neuronal geometry is reduced to a one-dimensional structure of line segments, and the evolution of the membrane potential in time is given by a one-dimensional PDE – the *cable equation* – which incorporates the Hodgkin-Huxley model of membrane currents in each compartment. Two well-known simulators for these kinds of models are NEURON [53] and GENESIS [23]. There is a vast number of studies using cable equation models, and it is impossible to even list the most important ones¹.

Several approximations and assumptions are made by the cable equation, which are explained in detail in [56]. These assumptions are briefly summarized below.

1. The neuron cross section is small compared to its length, so variations in radial direction can be neglected; the neuron can be accurately represented as a 1D structure.
2. The extracellular space is isopotential (grounded) and does not have a reverse effect on the neuron.
3. Magnetic fields are negligible.
4. All quantities, particularly membrane currents, can be described by continuous variables, i.e. stochastic effects can be neglected, since the number of participating ions is large enough and the ion diameter is sufficiently small compared to the relevant physical length scales.
5. Intra- and extracellular spaces can be represented as a homogeneous medium with effective physical parameters, e.g. diffusion coefficients and resistivity.

¹As of March 14 2014, Google Scholar showed over 1400 citations of the original NEURON paper by Hines and Carnevale.

6. Ion concentrations do not vary significantly, so they can be regarded as being constant.
7. The activity of a neuron is completely determined by its synaptic inputs and intrinsic states, i.e. the electric activity of other neurons in the vicinity does not have a direct effect.
8. The neuronal membrane can be modeled as a capacitor in parallel with a conductance.

This is quite a number of assumptions. While some are certainly justified, others can be questioned. In this context, particularly the assumptions of an isopotential extracellular space and its independence of the intracellular space as well as neglecting any concentration changes and their effects on the electric potential may be seen critically. There have been some interesting modeling efforts that address one or more of these approximations in order to come up with a more accurate model.

In [83], concentration dynamics were included by coupling the Nernst-Planck equation to the cable equation. Intracellular concentrations were allowed to change, while extracellular concentrations were kept constant. The authors reported deviations from the cable equation in small structures like dendritic spines, since in restricted volumes concentrations can vary significantly. Furthermore, an adjustment of the cable equation was proposed, where the constant ionic batteries in the Hodgkin-Huxley model were replaced by concentration-dependent Nernst potentials, which can be calculated from a separate set of equations for the ionic concentrations. In a second step, also the single constant intracellular resistances in each compartment were replaced by parallel resistances for each ion species, which led to a better agreement with the Nernst-Planck cable model.

One central problem of the cable equation is the exclusion of the extracellular space, which is the region of interest in this work. Most models for the extracellular potential are based on [volume conductor \(VC\) theory](#) [81], where the extracellular space is assumed to be electroneutral (and in most cases also homogeneous), i.e. any concentration effects by ionic charges are neglected. The relevant parameter for the extracellular medium in these models is the conductivity σ (or equivalently, its inverse, the resistivity ρ). The problem is reduced to the solution of the electrostatic part of Maxwell's equations, where the membrane is the only current source. The local changes in ion concentrations caused by drift and diffusion and their contributions to the electric field are not considered.

Instead, current sources are imposed as boundary conditions. This way, the electric potential can be calculated by Laplace's equation. A special case is the

line source approximation (LSA) [56], where the membrane surface is collapsed to a line source, for which an analytical solution can be expressed in cylinder coordinates. This avoids the need for a numerical solution and is computationally tractable, since one only has to compute it at the points of interest. It has also shown to give quite accurate results at distances larger than 1 μm from the membrane in an experimental comparison [45].

Refined models consider an inhomogeneous extracellular space [20] that account for effects like frequency filtering [19, 18].

An interesting technique in this context is the application of inverse methods to these kinds of models, enabling the estimation of current source densities from LFP measurements [80].

Another aspect of volume conductor models is that they are one-way models which take their inputs from a cable equation simulation, but the result is not fed back into the cable equation, such that intra- and extracellular space are still completely decoupled. In a recent work [2], this is accounted for by solving the governing equations not only on the extracellular domain, but on both intra- and extracellular domains, coupled by Hodgkin-Huxley-type interface conditions.

More detailed models explicitly consider the dynamics of ionic concentrations and their effects on the electric field, based on the Poisson-Nernst-Planck (PNP) system of electrodiffusion. These models use the ion concentrations of all participating species – next to the electric potential – as primary variables, i.e. they are explicitly solved for in the numerical simulations. In contrast to the work in [83] mentioned above, concentrations and electric potential are fully coupled through the Nernst-Planck and Poisson equations, respectively. These models provide the most fundamental representation of neuronal dynamics among all continuum models, given that magnetic fields can be neglected, which is the case under physiological conditions² [87]. However, they impose a serious challenge on the implementation, as they have to be solved numerically using problem-specific methods.

In [74], the reason for the high computational demand of electrodiffusion models based on the PNP equations is discussed: the presence of a thin *Debye layer* close to the membrane – over which concentrations change significantly – necessitates a very fine spatial resolution. A clever approximation is suggested which replaces the Debye layer by a special boundary condition, such that the remaining parts of the domain can be approximated as electroneutral, eliminating the concentrations from the equations. An extensive explication of the underlying asymptotic studies

²According to [87], the inclusion of magnetic fields into the calculation makes a difference of about one in 10^9 , as the relevant frequencies in neuronal tissue are small.

can be found in [73]. An interesting by-product of this remarkable study is the demonstration of the existence of a larger “intermediate layer” between Debye layer and electroneutral regions. In [75], this methodology was applied to study the effect of ephaptic coupling on cardiac action potential propagation in the absence of any synapses or gap junctions. Many more studies are specifically concerned with ephaptic coupling between neighboring cells, as reviewed in [40, 59]. We will defer the detailed literature review of this particular topic to chapter 7.

In [68], a detailed 3D numerical simulation of electrodiffusion based on the PNP equations was carried out for one node of Ranvier, showing the accumulation and depletion of ions close to the membrane, and therefore the invalidity of the electroneutrality approximation close to the membrane, as used in cable equation and VC models. However, the study focused on deviations from the cable equation, not on the extracellular signal, and the membrane thickness was greatly overestimated as a consequence of the coarseness of the spatial discretization dictated by the available computational resources. This also means the Debye layer was not accounted for. Although these limitations render quantitative statements difficult, it is the only study we know of dealing with the full set of PNP equations in the setting of neuronal excitation.

In other fields, particularly in biophysical studies of ion channels, the application of PNP is well-established. A review is given in [30]. Numerical methods for electrodiffusion-reaction equations were analyzed in a comprehensive way in [69], with special regard to surface potentials of biomolecules.

In summary, so far no model exists that explicitly resolves the relevant spatial scales to study neuronal excitation on the detailed level of electrodiffusion. Most importantly, there is no study we know of that actually uses the electrodiffusion approach to model the extracellular potential.

1.4 Outline

In this work we strive to model the detailed evolution of the concentrations of the most relevant ion species and the resulting electric field inside and (particularly) outside the cell during the spread of an action potential along an axonal membrane.

To this end, the Poisson-Nernst-Planck equations are solved numerically by application of the [finite element method \(FEM\)](#). To cope with the computational demand, a cylinder symmetry is introduced, which enables us to solve a 3D problem at 2D costs. We propose an efficient numerical scheme, particularly a suitable computational grid which resolves the multiple spatial scales accurately while still using only a minimal number of unknowns.

The outline of this thesis is as follows: in the following chapter, the relevant theoretical background will be treated. Chapter 3 discusses the general model, its prominent features and the numerical methods used to solve the system. A description of the implementation of the numerical algorithm follows in chapter 4. In chapter 5, the results for an unmyelinated axon in extracellular fluid are shown. We try to elucidate the evolution of the extracellular signal by looking at different contributing mechanisms and compare the results with VC models. In chapter 6, the modeling approach is extended to also include myelination. The necessary modifications in the numerical scheme and the emerging simulation results are discussed. In chapter 7, we consider neighboring nerve fibers and their ephaptic effects on each other. Chapter 8 gives a summary and discusses the results and future directions.

Some material in this work has been previously published in [82], parts of which are reproduced and cited in the following without explicit mention. This applies to section 1.3 and large parts of chapters 3, 5, and 8.

2 Theory

Very simple was my explanation, and plausible enough – as most wrong theories are!

(H.G. Wells)

This chapter serves as a basic outline of the theoretical background of this work. It is divided into two parts: a mathematical part concerned mainly with the numerical solution of [PDEs](#) and a second part that tries to list some fundamental biophysical relations in the context of neuroscience.

2.1 Mathematical Aspects of Partial Differential Equations

This section aims to act as a mini-primer on [PDEs](#) and its numerical solution by the [FEM](#). We will exclude the treatment of [ordinary differential equations \(ODEs\)](#) here and refer to the well-established theory [49, 50] instead.

After a short look on the types of equations and their most important representatives, we will shortly introduce the basic concept of the finite element method. By no means is this a complete treatise in the mathematical sense, for this we refer to the excellent script [14], which we will be largely following hereafter.

2.1.1 Type Classification

A general linear second-order [PDE](#) can – assuming continuously differentiable coefficients – be written in the form

$$Lu = - \sum_{i,j=1}^n a_{ij}(x) \partial_{x_j} \partial_{x_i} u + \sum_{i=1}^n b_i(x) \partial_{x_i} u + c(x)u = f \quad \text{in } \Omega \quad (2.1)$$

in the unknown u , where ∂_{x_i} is the shorthand notation for the partial derivative $\frac{\partial}{\partial x_i}$ with respect to x_i and Ω is some domain, usually a subset of the n -dimensional Cartesian space \mathbb{R}^n . L is called a linear *differential operator*.

Defining the matrix $(A(x))_{ij} = a_{ij}(x)$ and the vector $b(x) = (b_1(x), \dots, b_n(x))^T$, eq. (2.1) is said to be

- *elliptic* in x , if all eigenvalues of $A(x)$ are nonzero and have the same sign,
- *hyperbolic* in x , if all eigenvalues are nonzero, $n - 1$ eigenvalues have the same sign and the remaining eigenvalue has the opposite sign, or
- *parabolic* in x , if one eigenvalue is zero, the remaining eigenvalues have the same sign and the $n \times (n + 1)$ matrix $(A(x), b(x))$ has full rank.

If this classification holds for every $x \in \Omega$, this local property becomes a global one and eq. (2.1) is simply called elliptic (parabolic, hyperbolic).

The names stem from the case $n = 2$, where the level set of equal function values $q = \text{const.}$ of the quadratic form $q(x_1, x_2) = a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2$ takes either the shape of an ellipse, a parabola, or a hyperbola, depending on the values of the coefficients a_{ij} . Note that this classification is not complete, i.e. there are also PDEs of *mixed type*.

This classification of PDEs is useful for their numerical solution. There does not exist a general theory on the solution of partial differential equations, but for the subclasses, statements can be made about existence and uniqueness of solutions as well as about some of the properties these solutions have. This can be exploited to develop numerical methods tailored to equations in a certain class.

2.1.2 Prototypes

We will now have a look at some typical representatives of the above classes. In the following, $\Omega \subset \mathbb{R}^n$ is a (spatial) domain, $\Sigma = (0, T]$ is a time interval and $u : \Omega \rightarrow \mathbb{R}$ or $u : \Omega \times \Sigma \rightarrow \mathbb{R}$ denotes the unknown scalar function. We will also specify boundary and initial conditions, which will have to fulfill additional criteria (cf. [14, chapter 6]) in order to obtain a well-posed problem.

2.1.2.1 The Poisson Equation

The Poisson equation is a second-order elliptic PDE that arises naturally from many physical problems. It can for instance be used to calculate the gravitational potential due to certain masses, or the electrostatic potential due to a given

charge density. It reads

$$\begin{aligned} \Delta u &= f && \text{in } \Omega \\ u &= g && \text{on } \Gamma_D \\ \nabla u \cdot \mathbf{n} &= j && \text{on } \Gamma_N \end{aligned} \quad (2.2)$$

with *Dirichlet* boundary conditions g and *Neumann* boundary conditions j on the boundary $\partial\Omega = \Gamma_D \cup \Gamma_N, \Gamma_D \cap \Gamma_N = \emptyset$. Here and in the following, \mathbf{n} denotes the unit outer normal vector. It is important to note that, since the Poisson equation is not depending on time, it describes an *instantaneous* response or a *stationary* state at which the system will settle given the source term f . If $f = 0$, eq. (2.2) reduces to the *Laplace equation*.

2.1.2.2 The Heat Equation

The Heat equation is a second-order parabolic [PDE](#) describing the distribution of a physical quantity (e.g. heat) as a diffusive process in a certain region over time:

$$\begin{aligned} \partial_t u - \Delta u &= f && \text{in } \Omega \times \Sigma \\ u &= g && \text{on } \Gamma_D \times \Sigma \\ -\nabla u \cdot \mathbf{n} &= j && \text{on } \Gamma_N \times \Sigma \\ u &= u_0 && \text{on } \Omega \times \{0\}. \end{aligned} \quad (2.3)$$

with Dirichlet values g , Neumann values j and the *initial condition* u_0 . In contrast to the Poisson equation, the heat equation is *instationary*, as it contains a derivative with respect to time, meaning that it describes a *transient* process.

2.1.2.3 The Transport Equation

The transport or *advection* equation is a first-order hyperbolic [PDE](#) reading

$$\begin{aligned} \partial_t u + \nabla \cdot (vu) &= f && \text{in } \Omega \times \Sigma \\ u &= g && \text{on } \Gamma_D \times \Sigma \\ vu \cdot \mathbf{n} &= j && \text{on } \Gamma_N \times \Sigma \\ u &= u_0 && \text{on } \Omega \times \{0\}. \end{aligned} \quad (2.4)$$

with Dirichlet, Neumann and initial conditions as before. It may for instance describe the transport of certain substances in a fluid due to a known velocity field v .

2.1.2.4 The Convection-Diffusion Equation

The convection-diffusion equation is a second-order mixed-type PDE, as it is a combination of the parabolic heat eq. (2.3) and the hyperbolic transport eq. (2.4). It describes the distribution of the quantity u due to diffusion and advection (convection) over time:

$$\begin{aligned} \partial_t u - \nabla \cdot (D \nabla u) + \nabla \cdot (vu) &= f && \text{in } \Omega \times \Sigma && (2.5) \\ u &= g && \text{on } \Gamma_D \times \Sigma \\ (-D \nabla u + vu) \cdot \mathbf{n} &= j && \text{on } \Gamma_N \times \Sigma \\ u &= u_0 && \text{on } \Omega \times \{0\}. \end{aligned}$$

Here, D denotes the *diffusion coefficient*, which might in general be tensorial. In the following, we assume the diffusion to be isotropic, such that D is a scalar value.

The ratio between advective and diffusive forces can be quantified by the *Péclet number*

$$\text{Pe}_L = \frac{Lv}{D} \quad (2.6)$$

for a characteristic length L .

The bored reader at this point might wonder why the previous prototypic PDEs were introduced out of any apparent context. We beg for patience until the governing equations of electrodiffusion are introduced at the end of this chapter. It will then become clear that the PNP system is a combination of the above prototypes consisting of one elliptic and one convection-diffusion-type equation, i.e. it contains parts of all of the prototypes introduced above.

2.2 Numerical Solution of PDEs by the Finite Element Method

The task of solving PDEs like those in section 2.1.2 is really that of finding a *function* $u(x)$ that satisfies the PDE at every point in $\Omega \times \Sigma$. As there is an (uncountable) infinite number of functions to choose from, this is certainly not an easy task. When there is no analytical solution at hand, the goal is to find a *numerical* (approximate) solution by reducing the solution candidates to a finite number by *discretization*. This is the process of finding a finite number of points

$x_i \in \Omega, i = 0, \dots, N_S - 1$ (in space) and $t_i \in \Sigma, i = 0, \dots, N_T - 1$ (in time) on which the solution is to be satisfied.

The common choice for instationary equations is the *method of lines*, where one first discretizes the spatial variables, leaving the time continuous, and afterwards the time variable. The last step reduces the task to the solution of an ODE in only one variable, for which a suitable time-stepping method can be used. Both discretizations yield a (spatial/temporal) *grid* on which we seek the discrete solution of the continuous problem. For the space discretization, the FEM is the method of choice in the following.

2.2.1 Triangulation of the Domain

The finite element method was designed to find a solution to a given PDE in function space. For this, the spatial domain Ω is discretized to yield a *triangulation* (also *mesh*, *grid*) $\mathcal{T} = \{t_j : j = 0, \dots, m - 1\}$ consisting of *elements* t_j . In one dimension, the domain $\Omega = (a, b)$ is subdivided into

$$a = x_0 < x_1 < \dots < x_m = b$$

(not necessarily equidistant). Then for $j = 0, \dots, m - 1$, the elements t_j and the *mesh size* $h(t_j)$ are given by

$$t_j = (x_j, x_{j+1}) \text{ and } h(t_j) = x_{j+1} - x_j .$$

We use $h = \max_{t \in \mathcal{T}} h(t)$ to denote the maximum mesh size.

In higher dimensions, common choices for the elements are *simplices* (2D: triangles; 3D: tetrahedra) or *cuboids* (2D: rectangles, quadrilaterals; 3D: cubes/cuboids, hexahedra). Finding a suitable mesh is a problem on its own, as a “good grid” depends on the considered equation and its physical properties, as we will see later. Here, we will not delve into the details of mesh generation or adaptation and simply assume that a suitable triangulation \mathcal{T} has already been found.

2.2.2 The Finite Element Method

2.2.2.1 Weak Form

Before introducing Galerkin’s method, the heart of the finite element method, we need to spend a few words on the requirements for the solution candidate functions $u_h \in V_h$ living on the triangulation \mathcal{T} .

Take for example the Poisson eq. (2.2): since it is of second order, a function that satisfies the equation must be at least twice continuously differentiable. This

imposes a severe restriction on the underlying function space V_h , as in practice one might be perfectly content with a piecewise linear solution, which is only once piecewise differentiable. It is therefore customary to solve the *weak formulation* of the original PDE, which is obtained by multiplying the equation by a suitable *test function* (the “variation”) and integrating over the domain.

For the exemplary eq. (2.2) with homogeneous Dirichlet conditions $g = 0$ on $\Gamma_D = \partial\Omega$ ($\Gamma_N = \emptyset$), we choose v to be vanishing at the Dirichlet boundary, yielding

$$\int_{\Omega} \Delta u v \, dx = \int_{\Omega} f v \, dx . \quad (2.7)$$

After integrating by parts, the weak form reads

$$\int_{\Omega} \nabla \cdot (\nabla u) v \, dx = - \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx , \quad (2.8)$$

where we utilized that the boundary term vanishes due to $v = 0$ on $\partial\Omega$. Defining the *bilinear* and *linear forms*

$$a(u, v) = - \int_{\Omega} \nabla u \cdot \nabla v \, dx \quad \text{and} \quad l(v) = \int_{\Omega} f v \, dx ,$$

the problem now takes the form

$$\text{Find } u \in U : \quad a(u, v) = l(v) \quad \forall v \in V. \quad (2.9)$$

A solution u to eq. (2.9) is called a *weak solution*. The weak solution is also a solution of the *strong formulation* eq. (2.2) if u is in $C^2(\Omega)$, i.e. if it is twice continuously differentiable on the given domain (see [39, chapter 8.1.2]). Note that even though we used a concrete example above, every linear PDE can be written in the form eq. (2.9).

We now quickly touch the concept of *linear operators*. Let $\mathcal{L}(V, \mathbb{R})$ denote the set of all *linear and continuous operators* A from V to \mathbb{R} . $V' = \mathcal{L}(V; \mathbb{R})$ is called the *dual space* to V . This way, we obtain an alternative notation of eq. (2.9), the operator notation:

$$\text{Find } u \in U : \quad Au = l. \quad (2.10)$$

Here, we have used the shortcut notation $Au = A(u)$ and the definition

$$\forall u \in U, \forall v \in V : \langle Au, v \rangle_{V', V} = a(u, v) ,$$

so $Au \in V'$ is the continuous linear form that is obtained from $a(u, v)$ by fixing the argument u . The scalar product $\langle \cdot \rangle_{V', V}$ is defined according to the *Riesz Representation Theorem* [14, Theorem 5.13].

Thus, eq. (2.9) has a unique solution if and only if the corresponding operator A is *invertible*, i.e. if it is injective and surjective. This problem formulation nicely reflects the practical implementation, where A is the matrix, u is the coefficient vector representing the solution, and l is the right-hand side of the linear system to be solved.

In summary, the finite element method relies on the solution of PDEs in weak form, as it requires less *regularity* (“numbers of derivatives”) of the candidate functions, allowing for instance the choice of piecewise linear finite element functions for the solution of a second-order problem. This greater freedom in controlling both accuracy and computational efficiency by the choice of a less regular finite element functions has turned out to be very beneficial in practice.

2.2.2.2 Existence and Uniqueness of Weak Solutions

The weak formulation eq. (2.9) of a given PDE obviously has practical advantages over the strong formulation. But is it also well-posed, i.e. does it have a unique solution? To answer this question, we have to become a little more precise and specify the classes of function spaces V we will be using and their different properties. In the following, we always assume the spaces to be defined on some domain Ω over the field of real numbers \mathbb{R} . For a formal definition of these important spaces, see [14, chapter 5].

- **Banach space:** A vector space equipped with a *norm* $\|\cdot\|$ which is *complete* with respect to this norm, i.e. every Cauchy sequence will converge to an element in the space V .
- **Hilbert space:** A Banach space which is additionally equipped with a scalar product $\langle u, v \rangle$ and an induced norm $\|u\| = \sqrt{\langle u, u \rangle}$.
- **Lebesgue space L^2 :** A Hilbert space whose elements are required to be square-integrable, i.e. $\int_{\Omega} |u(x)|^2 dx < \infty$; here we see the connection to integrals of the form eq. (2.7). This property also allows for the definition of the scalar product $\langle u, v \rangle = \int_{\Omega} u(x)v(x) dx$ and the induced L^2 norm $\|u\| = \sqrt{\langle u, u \rangle} = \left(\int_{\Omega} u(x)^2 dx\right)^{\frac{1}{2}}$.
- **Sobolev space $H^k \subset L^2$:** A subset of L^2 which additionally requires the existence of *weak derivatives* up to order k ; a weak derivative $\partial_x u$ of a

function u satisfies $\langle u, \partial_x v \rangle = -\langle \partial_x u, v \rangle$ for every test function $v \in C_0^\infty$. This ensures the existence of integrals of the form of those in eq. (2.8) obtained by integration by parts. The scalar product is given by $\langle u, v \rangle = \sum_{0 \leq |\alpha| \leq k} \int_\Omega (\partial^\alpha u)(\partial^\alpha v) dx$ and the induced norm as before by $\|u\| = \sqrt{\langle u, u \rangle}$.

One commonly used example is the Sobolev space H^1 of functions with weak first order derivatives and scalar product $\langle u, v \rangle = \int_\Omega uv + \nabla u \cdot \nabla v dx$ inducing the H^1 norm $\|u\| = (\int_\Omega u^2 + \|\nabla u\|^2 dx)^{\frac{1}{2}}$.

Later on, we will see two concrete choices of function spaces for this rather abstract setting: the space of piecewise polynomials on a given simplicial or cuboid mesh, which is also used in the classical **FEM**.

The Lax-Milgram theorem (cf. [14, Theorem 5.21]) ensures the existence of a unique solution of the weak problem eq. (2.9).

Theorem 1 (Lax-Milgram). *Let V be a Hilbert space, $a \in \mathcal{L}(V \times V; \mathbb{R})$ and $l \in V'$. Then Problem eq. (2.9) is well-posed provided a is coercive (or elliptic), i.e. it satisfies the condition*

$$\exists \alpha > 0, \forall u \in V : \quad a(u, u) \geq \alpha \|u\|^2.$$

Moreover, the following *a-priori* estimate holds:

$$\forall l \in V' : \quad \|u\|_V \leq \frac{1}{\alpha} \|l\|_{V'}.$$

A direct proof can be found in [24]. An alternative is to reduce the Lax-Milgram theorem to the Banach-Nečas-Babuška theorem [14, Theorem 5.2.1], which is more general and only requires V to be a Banach space, not necessarily a Hilbert space. The ellipticity of a implies that the corresponding operator A in eq. (2.10) is invertible.

2.2.2.3 The Continuous Galerkin Method

To obtain finite-dimensional functions for the representation of solutions, we need a corresponding function space V_h . The subscript h of V_h refers to the mesh size, indicating that it is a finite subspace living on a certain mesh. An appropriate choice of V_h will satisfy

$$\inf_{u_h \in V_h} \|u - u_h\| \rightarrow 0 \quad \text{as } h \rightarrow 0,$$

such that for small mesh sizes h , u_h comes arbitrarily close to the function u living on the original space V . We then pick *trial* or *ansatz functions* which form a *basis* of the underlying function space V_h , such that every function u_h can be represented as a linear combination of the ansatz functions.

There are two principal variants for the choice of ansatz functions: in the **continuous Galerkin (cG)** method they are defined to be globally continuous, i.e. there are no jumps at element boundaries. The **discontinuous Galerkin (dG)** method, which we will not cover here, allows the ansatz functions to be discontinuous at element boundaries, making them especially useful for **PDEs** which themselves have discontinuous solutions, like the transport eq. (2.4) with an initial step condition.

Both variants need a second function space W_h for the test functions. In the classical Galerkin method, the spaces for ansatz and test functions are the same, $V_h = W_h$. If $V_h \neq W_h$, we obtain the class of *Petrov-Galerkin* methods, which will not be covered here.

Given a mesh \mathcal{T} , the abstract **cG** method now reads as follows:

1. Obtain the *weak formulation* of the **PDE** by integration and multiplication with test functions v .
2. Choose a suitable function space V_h for the ansatz functions on \mathcal{T} , e.g. the space of piecewise polynomials of degree k , $P_k(\mathcal{T}) / Q_k(\mathcal{T})$.
3. With the *bilinear form* $a(u, v)$ and *linear form* $l(v)$ from the weak formulation, state the problem in the finite-dimensional subspace as

$$\text{Find } u_h \in V_h : \quad a(u_h, v) = l(v) \quad \forall v \in V_h. \quad (2.11)$$

4. Find a basis of V_h

$$\Phi_h = \{\varphi_1^h, \dots, \varphi_{N_h}^h\}$$

of size $N_h = \dim V_h$.

5. As all function $v_h \in V_h$ can be represented as linear combinations of elements of Φ_h , it is sufficient to use these instead of all test functions v in eq. (2.11). Utilizing this and inserting the basis representation

$$u_h = \sum_{j=1}^{N_h} z_j \varphi_j^h$$

yields a linear system of equations

$$\begin{aligned} & a(u_h, v) = l(v) \quad \forall v \in V_h \\ \Leftrightarrow & a \left(\sum_{j=1}^{N_h} z_j \varphi_j^h, \varphi_i^h \right) = l(\varphi_i^h) \quad \forall i = 1, \dots, N_h \\ \Leftrightarrow & \sum_{j=1}^{N_h} z_j a(\varphi_j^h, \varphi_i^h) = l(\varphi_i^h) \\ \Leftrightarrow & Az = b \quad (A)_{ij} = a(\varphi_j^h, \varphi_i^h), \quad (b)_i = l(\varphi_i^h) \end{aligned}$$

for the coefficient vector $z \in \mathbb{R}^{N_h}$.

6. Solve the linear system by an appropriate linear solver.

Note how the discrete version eq. (2.11) of the weak problem formulation eq. (2.9) has been reduced to a linear system of equations, representing a discrete version of the operator formulation eq. (2.10).

2.2.2.4 Galerkin Orthogonality

We have just seen that Galerkin's method can be used successfully to solve a linear PDE in a finite-dimensional subspace. The question remains: why is this a good strategy? In particular, what does the multiplication with test functions and integration over the domain mean for this problem?

To understand this, let us again have a look at our Poisson example eq. (2.2). The residual for this equation reads $R(u) = \Delta u - f$, which is to be minimized by the Galerkin method. Rearranging eq. (2.7) yields

$$\int_{\Omega} (\Delta u - f)v \, dx = \int_{\Omega} R(u)v \, dx = 0. \quad (2.12)$$

Fulfilling this relation with a discrete function u_h for every $v \in V_h$ means that the residual is *orthogonal* to every test function v , as can be seen when expressing eq. (2.12) in terms of the L^2 scalar product¹,

$$\langle R(u_h), v \rangle = 0 \quad \forall v \in V_h. \quad (2.13)$$

¹This is completely analogous to the vector space of real numbers, where two vectors are orthogonal if their scalar product vanishes.

For the weak solution, eq. (2.12) takes the slightly different form

$$-\int_{\Omega} (\nabla u_h \cdot \nabla v + fv) dx = 0 \quad \forall v \in V_h, \quad (2.14)$$

or, more generally, for the abstract Galerkin problem eq. (2.11),

$$r(u_h, v) = a(u_h, v) - l(v) = 0 \quad \forall v \in V_h. \quad (2.15)$$

The approximation error $u - u_h$ therefore fulfills

$$\begin{aligned} r(u - u_h, v) &= r(u, v) - r(u_h, v) = a(u, v) - l(v) - (a(u_h, v) - l(v)) \\ &= a(u, v) - a(u_h, v) = a(u - u_h, v) = 0 \quad \forall v \in V_h. \end{aligned}$$

This property is called the *Galerkin orthogonality* of the error $u - u_h$ with respect to the bilinear form a . Since, after a choice of the space V_h , we are only interested in the errors that are actually *representable* in V_h , a solution whose error is orthogonal to all functions in the test space is optimal in this sense. The orthogonal part of the error can simply not be captured in V_h and therefore we have found the best solution available in V_h .

Another view on this problem is related to the L^2 norm $\|u\| = \sqrt{\langle u, u \rangle}$ induced by the scalar product. By demanding the orthogonality of the residual, Galerkin's method actually calculates the L^2 projection u_h of the function u into the finite-dimensional subspace V_h , which is known to give the best approximation in the L^2 norm [39, chapter 5.6].

2.2.2.5 Common Function Spaces

Now that we are convinced that Galerkin's method gives an optimal solution in a chosen function space V_h , we briefly mention two common choices for V_h for simplicial and cuboid grids.

Pk Finite Elements The standard choice of ansatz and test functions are polynomials. If we denote the space of polynomials of degree at most k in n space dimensions as

$$\mathbb{P}_k^n = \{u \in C^\infty(\mathbb{R}^n) : u(x) = \sum_{0 \leq |\alpha| \leq k} c_\alpha x^\alpha\}, \quad (2.16)$$

we can use the finite element space

$$P_k(\mathcal{T}) = \{u \in C^0(\bar{\Omega}) : u|_{\bar{t}} \in \mathbb{P}_k^n \forall t \in \mathcal{T}\}. \quad (2.17)$$

on a conforming simplicial mesh (e.g. triangles, tetrahedra).

Qk Finite Elements For cuboid meshes (rectangles, cubes/hexahedra), the space of polynomials of degree at most k is defined slightly differently as

$$\mathbb{Q}_k^n = \{u \in C^\infty(\mathbb{R}^n) : u(x) = \sum_{0 \leq |\alpha|_\infty \leq k} c_\alpha x^\alpha\} \quad (2.18)$$

with $|\alpha|_\infty = \max_i \alpha_i$, $\alpha_i \in \mathbb{N}_0$ and $\dim \mathbb{Q}_k^n = (k+1)^n$. If all cubes in the mesh \mathcal{T} are axi-parallel, the space

$$Q_k(\mathcal{T}) = \{u \in C^0(\bar{\Omega}) : u|_{\bar{t}} \in \mathbb{Q}_k \forall t \in \mathcal{T}\} \quad (2.19)$$

can be defined analogous to the simplex meshes. This finite element space will be used throughout this document with the choice of $k = 1$, i.e. it is a subspace of the Sobolev space H^1 . The method of finding a suitable basis for this space is covered in [14, chapter 7.5].

2.2.3 Nonlinear PDEs: Newton's Method

So far, we have only considered linear **PDEs**. But in practice, one often has to deal with nonlinear equations. Since the abstract existence theorems (Lax-Milgram, Banach-Nečas-Babuška) only hold for linear **PDEs**, the well-posedness of a nonlinear **PDE** has to be shown by other means, often on a case-by-case basis. There does not exist a general theory for the solvability of general **PDEs**, and it is commonly believed unlikely that such a general theory exists.

Assuming a unique solution exists, the generalization of the numerical solution of such problems by the **FEM** is quite straightforward, as the nonlinear operator can be linearized by *Newton's method* and embedded into an iterative procedure. Here we use the *damped Newton method*, see e.g. [34].

Given the initial guess u^0 , compute $r^0 = R(u^0)$. Set $k = 0$ and iterate until convergence:

1. Compute Jacobian matrix $A^k = \nabla R(u^k)$.
2. Solve $A^k z^k = r^k$ with some linear solver.

3. Update $u^{k+1} = u^k - \sigma^k z^k$, $\sigma^k \in (0, 1]$.
4. Compute the new residual $r^{k+1} = R(u^{k+1})$.
5. Set $k = k + 1$.

The residual $R(u)$ contains entries $a(u_h, v_h) - l(v_h)$ from our weak form Galerkin approximation in residual form. Additionally, the Jacobian $\nabla R(u)$, the derivative of the residual with respect to all elements of u , is needed. This can either be obtained by providing an analytical derivative of the operator L beforehand or by using numerical differentiation. For the choice of the damping factor σ , one can for instance use a *line search* strategy that tries to minimize the residual along the search direction z .

Note that for a linear operator L , $R(u) = Au - b$ and thus $\nabla R(u) = A$. In this case, the Newton method will simply solve the linear system and converge in one iteration.

2.2.4 Time Discretization

The theory of ODEs [49, 50] has produced a vast amount of time-stepping schemes, which is impossible to cover here. Instead, we will look at the Euler methods as representatives of the two major classes of methods.

Suppose the initial value problem to be solved is

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0 ,$$

where we can think of $y(t)$ as our solution of the FEM solution of our PDE at a certain time t . Suppose now as above the time interval has been discretized at N_T time points $t_i \in \Sigma, i = 0, \dots, N_T - 1$ with time intervals $\Delta t_i = t_{i+1} - t_i, i = 0, \dots, N_T - 2$.

The *explicit Euler* (or forward Euler) reads

$$y_{i+1} = y_i + \Delta t_i f(t_i, y_i), \quad i = 0, \dots, N_T - 2 . \quad (2.20)$$

It is a representative of the *explicit* time-stepping methods, as the right-hand side only depends on quantities from the previous iterations.

Its antagonist is the *implicit Euler* (or backward Euler) method

$$y_{i+1} = y_i + \Delta t_i f(t_{i+1}, y_{i+1}), \quad i = 0, \dots, N_T - 2 , \quad (2.21)$$

representing an *implicit* method, as the right-hand side depends on values that are not known yet. In general, implicit schemes require the solution of a linear

system of equations in each time step and hence tend to be more expensive. However, they possess better *stability* properties that in many cases allow the usage of larger time steps Δt_i .

Both methods in eqs. (2.20) and (2.21) are of *first order accuracy*, meaning that the error of this method scales with $\mathcal{O}(\Delta t_i)$.

2.2.5 Linear Solvers

As seen in the previous sections, the numerical solution of a PDE in the end is always reduced to the solution of a linear system of equations. The variety of methods for this kind of problem is overwhelming, ranging from the direct solution by Gauß elimination with a complexity of $\mathcal{O}(n^3)$ to very sophisticated sparse iterative solvers [88] and parallel multigrid methods [10] with an optimal complexity of $\mathcal{O}(n)$. A comprehensive list of methods is beyond the scope of this thesis, and since the choice of a suitable solver is highly problem-specific, we will defer this topic to the relevant chapters.

2.3 Fundamentals of Biophysics and Neuroscience

2.3.1 Structure and Function of Neurons

A neuron – see fig. 2.1 for an illustration – can be structurally divided into three parts: The *soma* containing the nucleus as the center of a neuron is the largest part. Multiple *dendrites* branch off from the soma, creating the *dendritic tree*, which receives its input at *synapses*, the connections to other neurons. The third part is the *axon*, which is a fiber that is often thinner than the dendrites, but nevertheless can show quite complex branching as well. A neuron always has one and only one axon.

Apart from this common property, axons can show a wide range of variations among neurons. They may or may not be *myelinated* as in fig. 2.1, i.e. insulated by a myelin sheath that is provided by a surrounding cell. In this case, the *nodes of Ranvier* provide periodical segments exposing the underlying axon directly to the *extracellular space (ES)*. The nodes usually contain a high density of ion channels, which provide for an active propagation of a traveling action potential by means of transmembrane currents that keep the AP alive.

Between the nodes, at myelinated parts (or *internodes*), the signal propagates passively, but significantly faster than in unmyelinated fibers, since virtually no charge is lost across the myelinated membrane. The established view is that axons are one-way paths, i.e. signals are always propagated away from the cell soma

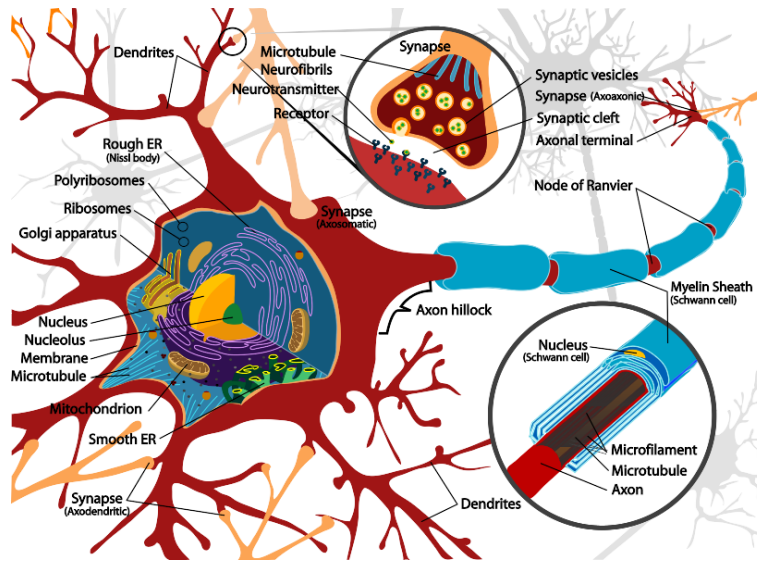


Figure 2.1: **Illustration of a typical neuron.** By [LadyofHats](#) [Public domain], via [Wikimedia Commons](#)

towards other cells in an *orthodromic* fashion, although there have been some hints that also *antidromic* propagation in the opposite direction may happen under physiological conditions [7].

Figuratively (and keeping with the author’s favorite metaphor of electronic devices), one can think of the dendrites as the neuron’s antennae collecting input from other neurons, while the axon is the output channel. Action potentials are generated at the *axon initial segment* close to the soma, and propagate down the axonal arbor, where they will eventually stimulate other neurons (see fig. 2.2).

Stimulations of this kind, however, do not always have to be *excitatory*, that is driving the target neuron to fire another action potential, but they may also have an *inhibitory* effect, i.e. preventing the target neuron from firing. This depends on whether the synapse transmitting the signal between two neurons – or more specifically, the *neurotransmitters* it uses as a messenger – has excitatory or inhibitory character. This is important, since an over-excitation can lead to pathological situations in the brain, one extreme example being epilepsy. The balance of excitation and inhibition is crucial for a functional brain.

The question remains how the elementary unit of information, the action potential, is generated. As mentioned above, the axon plays the leading role in this process. It turns out that nature takes advantage of elementary electrodynamics,

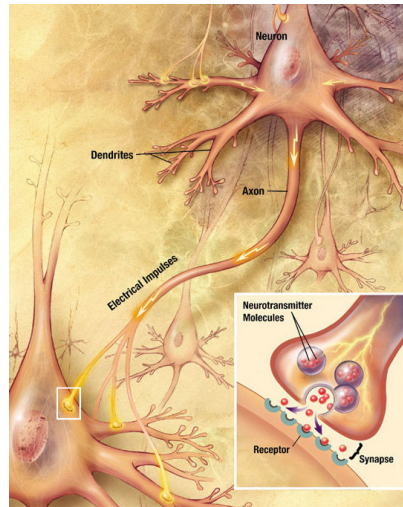


Figure 2.2: **Neuronal signaling mechanisms.** By [Looie496](#) created file, US National Institutes of Health, National Institute on Aging created original [Public domain], via [Wikimedia Commons](#)

using ions as charge carriers and the axonal membrane as an electric circuit. In the following, some basic properties of passive membranes will be touched, before we turn to the active parts of the membrane, the ion channels.

2.3.2 Membrane Physics

A biological membrane of finite thickness d_{memb} and an electric permittivity ϵ_{memb} has some important electric properties with influence on the two electrolytes it separates. In the following, we will only consider the case where the electrolytes on both sides of the membrane have the same solvent, water. Therefore, the electrolyte permittivity ϵ_{elec} is a single value, and the only difference between both sides is the ion concentration. For the two electrolytes, the terms *cytosol* and *extracellular space* will be used hereafter, often abbreviated as CY or ES in sub- or superscripts, respectively.

A schematic depiction of the membrane and one adjacent electrolyte is given in fig. 2.3. We see that the (electrically charged) membrane attracts oppositely charged counterions and repels equally charged co-ions. As a consequence, an **electric double layer (EDL)** forms: one layer of counterions directly at the membrane, and another layer of co-ions, which is attracted by the counterion layer. Several names are linked with the theory of electric double layers. Helmholtz

[51] was the first to realize that such an EDL has the capacity to store electric charges and therefore acts as a capacitor. Gouy and Chapman [46, 28] noted that this capacitance depends on the applied membrane voltage and the ionic concentrations; they were also the first to find that the ion concentrations decrease exponentially with distance from the membrane, which can be described by Maxwell-Boltzmann statistics. This added a *diffuse layer* to the Helmholtz layer of fixed charges directly at the membrane interface. Further improvements were made by Stern [93] and others, resulting in a quite complex theory of different layers and their interactions.

A very good summary of the theoretical background is given in [67, chapter 12]. One central equation describing the potential profile in the presence of a membrane is the Poisson-Boltzmann equation.

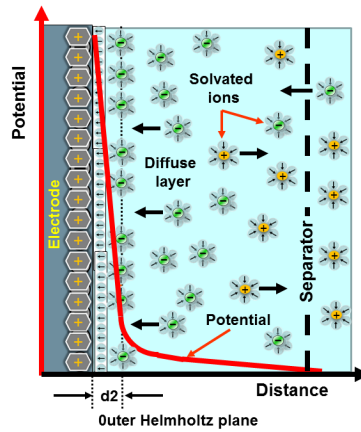


Figure 2.3: **Schematic picture of a membrane and electrolyte.** By Elcap (Own work) [CC0], via [Wikimedia Commons](#)

2.3.2.1 Poisson-Boltzmann Equation

The Poisson-Boltzmann equation

$$\nabla \cdot (\epsilon \nabla \phi) = -\frac{e^2 n^*}{\epsilon_0 k_B T} \sum_i z_i n_i^0 \exp(-z_i \phi) \quad (2.22)$$

is a special case of the Poisson equation. Here, ϕ is the dimensionless relative electric potential energy with respect to the thermal energy ($\phi = eU/k_B T$ with U

given in units of volts; at room temperature, $\phi = 1$ corresponds to $U \approx 25$ mV); ϵ is the relative permittivity, which again may be position-dependent, and T is the temperature of the solvent. The remaining constants are the vacuum permittivity ϵ_0 , the Boltzmann constant k_B , and the elementary charge e .

In comparison to eq. (2.2), the charge density on the right-hand side has been replaced by a Boltzmann distribution for the ion concentrations at equilibrium

$$n_i = n_i^0 \exp(-z_i \phi) \quad (2.23)$$

due to a certain potential that has been established at the membrane. Here, n_i denotes the concentration of species i with valence z_i and bulk concentration n_i^0 .

The equation basically describes how a charged membrane in an ionic solution causes accumulation of counterions, i.e. ions with opposite charge with respect to the membrane charge. This membrane charge could be some surface charge of a biomolecule or, as in this work, the charge due to ions on the other side of the membrane. In either case, the potential profile will cause a charge density profile with opposite sign, as counterions are attracted and co-ions are repelled from the membrane with a certain charge. As a consequence, the previously described EDL of two oppositely charged regions forms at the boundary between membrane and electrolyte. Figure 2.3 shows a typical potential profile as described by eq. (2.22).

The Poisson-Boltzmann equation proves useful to test a numerical algorithm, as the calculated equilibrium charge density can be compared with the analytical expression eq. (2.23).

2.3.2.2 Debye-Hückel Theory

For monovalent solutions and small surface potentials (< 25 mV according to [67]), eq. (2.22) can be linearized to the Debye-Hückel equation [32]

$$\nabla^2 \phi = \frac{1}{\gamma_D^2} \phi, \quad (2.24)$$

where

$$\gamma_D = \sqrt{\frac{\epsilon_{\text{elec}} \epsilon_0 k_B T}{e^2 2 I S}} \quad (2.25)$$

is called the *Debye length*. This is an important property of an electrolyte which is inversely proportional to the square root of the *ionic strength*

$$\text{IS} = \sum_i \frac{1}{2} n_i z_i^2 . \quad (2.26)$$

The Debye length is a characteristic spatial scale over which electrostatic interactions close to the membrane are *screened* (i.e., decaying) exponentially. It therefore gives a measure on the distance over which the membrane influences the electric field to a large degree.

For strong electrolytes, γ_D is small. However, for electrolytes of lower concentration, the Debye length is larger and has a significant effect in a larger range around the membrane. A common term for this vicinity is the *Debye layer*, the region of up to a few Debye lengths. Here, we will define the Debye layer to be 10 times the Debye length. Outside the Debye layer one can safely assume that concentrations have decayed to their bulk values, such that membrane effects do not play a role.

2.3.3 Ion Channels

It is obvious that a simple solid membrane will not be sufficient to yield some kind of excitation. After the previous demonstrations, we know that the Debye layer experiences an accumulation of ions towards the membrane. Hence, we can think of the membrane as a *capacitance* separating charges present on the two adjacent electrolytes. In order to allow charge carriers to actually cross the membrane, we now introduce *ion channels* in the membrane.

With this, the neuronal membrane can be adequately described by an equivalent circuit as in fig. 2.4, consisting of a capacitance in parallel with a conductance and a battery. The conductance now represents one out of the multitude of *ion channels* that exist in biological membranes, allowing for a current to establish between intra- and extracellular space. The battery here stands for the reversal potential of the ion channel. We will now have a look at the most important channel types responsible for the action potential generation.

2.3.3.1 Leak Channels

The simplest kind of ion channels are *leak channels*, which have a constant conductance g_L . These play an important role for the resting state of a neuron. The current flowing through channels of this type will depend on the concentration gradient ∇n_i and the potential gradient $\nabla \phi$ between both sides of the membrane

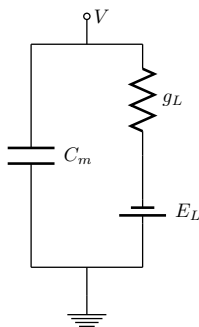


Figure 2.4: **Equivalent circuit for a patch of membrane.** A patch of membrane can be represented by a capacitance C_m in parallel with a series of a leak conductance g_L and a battery E_L representing the (Nernst) reversal potential.

(cf. [17]). For one ion species, *Nernst's Equation* is very successful in predicting the *reversal potential*

$$E = \frac{RT}{zF} \ln \frac{n^{\text{ES}}}{n^{\text{CY}}} \quad (2.27)$$

of a leak channel that is selective for a certain ion. Here, R denotes the *gas constant* and F the *Faraday constant*. The **Goldman-Hodgkin-Katz (GHK)** or simply *Goldman equation* extends this to the case of multiple monovalent species, reading

$$E = \frac{RT}{F} \ln \left(\frac{\sum_i^N P_{M_i^+} [M_i^+]^{\text{ES}} + \sum_j^M P_{A_j^-} [A_j^-]^{\text{CY}}}{\sum_i^N P_{M_i^+} [M_i^+]^{\text{CY}} + \sum_j^M P_{A_j^-} [A_j^-]^{\text{ES}}} \right), \quad (2.28)$$

with $[M_i^+]$ and $[A_j^-]$ denoting cation and anion concentrations, respectively, and P_i being the relative permeability of the membrane for ion species i . E gives the *resting potential* of a neuron with respect to extracellular space.

The Goldman equation makes some assumptions about the intrinsics of the ion channel: first of all, it assumes a constant electric field (i.e., a linear potential drop) over the membrane; secondly, all ions cross the membrane independently, so there is no interaction between them. It is clear that we need the (relative) permeabilities of the membrane for all considered ion species in order to predict the resting potential in this way. In practice, permeabilities are not as easily available

as conductances, although the quantities are related. However, a conductance as a purely electric property can readily be obtained from a direct membrane current measurement by Ohm's law, while this proves more difficult for permeabilities, which is a chemical property of the ions involving the diffusion coefficient and the water-membrane partition coefficient.

When dealing with conductances, one can instead use the *parallel conductance model (PCM)* [106], which simply uses Ohm's and Kirchhoff's laws applied to an equivalent circuit of the form in fig. 2.4 with multiple leak conductances in parallel, one for each ion species. Requiring the net membrane flux to be zero at equilibrium, we get the simple relation

$$E = \frac{\sum_i^N g_i E_i}{\sum_i^N g_i} \quad (2.29)$$

with the leak conductances g_i and corresponding reversal potentials E_i as calculated by eq. (2.27) for each ion species separately. It is easy to see that this reduces to the Nernst eq. (2.27) for a single species.

When dealing with equations that calculate the resting potential of neurons due to the concentrations of certain ions on both sides of the membrane, it is important to note that all of these are based on certain assumptions that might or might not be true in the specific case:

One might ask why the voltage equation is so hard to derive and so closely tied to minute assumptions when the analogous Nernst equation is so simple to obtain [...] and so general. The contrast is typical of the difference between equilibrium and nonequilibrium problems. The Nernst equation describes a true equilibrium situation and can therefore be derived from thermodynamics as a necessary relation between electrical and "concentration" free energies with no reference to structure or mechanism. On the other hand, the zero-current voltage equation represents a dissipative steady state. [...] Only the sum of charges moving is zero. The reversal potential is not a thermodynamic equilibrium potential. Such nonequilibrium problems often can make little use of thermodynamics and require empirical relationships closely tied to the structure and mechanism of the flow. The assumptions are often simplistic and no more than approximations. They are models. [52, p. 452]

With that in mind, it depends on the underlying model whether eq. (2.28) or eq. (2.29) is adequate to calculate the resting potential E . In practice, it also

depends on the availability of parameters, i.e. whether we think in terms of permeabilities or conductances. In the following, we will take the electrical engineer's point of view, and look at the membrane in terms of conductances and equivalent circuits.

Leak channels are also called *passive* channels, in contrast to *active* channels, whose conductance changes in time, depending on the membrane potential or the concentrations of certain compounds. We will restrict ourselves to the voltage-dependent ones, whose dynamic properties were described in the seminal work by Hodgkin and Huxley.

2.3.3.2 The Hodgkin-Huxley System of Membrane Excitation

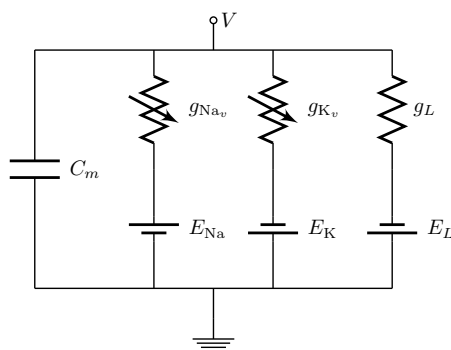


Figure 2.5: **Equivalent circuit for the Hodgkin-Huxley membrane patch model.** It consists of a membrane capacitance C_m in parallel with three branches, each consisting of a series of conductance and battery, representing the ionic current and reversal potentials, respectively. Note that the voltage-gated conductance are dynamic, while the leak conductance does not change over time. Reversal potentials of sodium (E_{Na}) and potassium (E_K) have opposite sign.

Hodgkin and Huxley [55] studied ion channel kinetics in the squid giant axon, a fiber that can be up to 1 mm in diameter, providing a very accessible way for electrophysiologic measurements. They considered only three ion channel types: two voltage-dependent channels selective for sodium (Na^+) and potassium (K^+), respectively, and one (virtual) leak channel. This results in the electric circuit of fig. 2.5 with three different conductances g_L , g_{Na_v} and g_{K_v} with associated currents I_L , I_{Na} and I_K . We largely follow [65] for the rest of this section.

Using Kirchhoff's current law for a parallel circuit, the equation for the current reads

$$C_m \frac{\partial V}{\partial t} = I_L + I_{\text{Na}} + I_{\text{K}} + I_{\text{inj}} , \quad (2.30)$$

where V is the membrane potential and C_m is the *membrane capacitance* responsible for the capacitive current. I_{inj} stands for the injected current or, more generally, any current source contributing to the potential on the intracellular side of the membrane. The ionic currents can be expanded to

$$I_L = g_L(E_L - V) \quad (2.31)$$

$$I_{\text{Na}} = g_{\text{Na}_v}(E_{\text{Na}} - V) \quad (2.32)$$

$$I_{\text{K}} = g_{\text{K}_v}(E_{\text{K}} - V) , \quad (2.33)$$

where E_i denotes the reversal potential of the ion channel i , as can be calculated by eq. (2.27). When the channel is permeable to more than one ion species – as the single leak channel above – its reversal potential is usually calculated using eq. (2.28) or eq. (2.29).

The constant leak conductance g_L is the inverse of the *membrane resistance*. It is very common to express the electric quantities in terms of unit length or unit area, since the measured values are often only available as *densities*. The membrane resistance then becomes a *resistivity* $r_L = R_L \pi r^2$ with the radius of the neurite r in cm, yielding the unit $\Omega \text{ cm}^2$. Analogously, the capacitance c_m per unit area in units F/cm^2 can be used instead of the total capacitance C_m .

The conductances above can also be expressed in terms of *conductivities* (commonly in units mS/cm^2), and the currents change to *current densities*, which need to be multiplied by the membrane area to get an absolute current.

The remaining active channel conductances $g_i(V, t)$ are defined as

$$g_{\text{Na}_v} = \bar{g}_{\text{Na}_v} m^3 h \quad (2.34a)$$

$$g_{\text{K}_v} = \bar{g}_{\text{K}_v} n^4 , \quad (2.34b)$$

where m, h, n are time- and voltage-dependent *gating particles* – taking values from the interval $[0, 1]$ – for the sodium activation, sodium inactivation, and potassium activation, respectively. In combination they state which fraction of the maximum conductance \bar{g}_i of channel i is open. It is notable that the concept of gating particles is a purely theoretical one, i.e. there is no direct physical equivalent in the chemical structure of an ion channel.

The kinetics of gating particles are given by the ODEs

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n \quad (2.35a)$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \quad (2.35b)$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h \quad (2.35c)$$

with corresponding rate functions

$$\alpha_n(V) = c_T 0.01 \text{vtrap}(10 - (V - V_{\text{rest}}), 10)$$

$$\beta_n(V) = c_T 0.125 e^{-(V - V_{\text{rest}})/80}$$

$$\alpha_m(V) = c_T 0.1 \text{vtrap}(25 - (V - V_{\text{rest}}), 10)$$

$$\beta_m(V) = c_T 4 e^{-(V - V_{\text{rest}})/18}$$

$$\alpha_h(V) = c_T 0.07 e^{-(V - V_{\text{rest}})/20}$$

$$\beta_h(V) = c_T \frac{1}{e^{(30 - (V - V_{\text{rest}}))/10} + 1},$$

which were slightly adapted to account for temperature dependence by a factor

$$c_T = 3^{\frac{T-6.3}{10}}$$

and zeros in the denominator of rate functions by a function `vtrap` as used in NEURON [27]:

$$\text{vtrap}(x, y) = \begin{cases} \frac{x}{e^{x/y} - 1} & x/y \neq 0 \\ y(1 - \frac{x}{2y}) & \text{else} . \end{cases}$$

This completes the description of the [Hodgkin-Huxley \(HH\)](#) model. It has since been modified to include arbitrary ion channel types in addition to the ones present in the original model by defining rate functions, gating particle kinetics and maximum conductances for each added channel, yielding one additional current in eq. (2.30). Adding dependence on the concentration of a certain ion or molecule is straightforward, given a good model for the time-evolution of concentrations is available. Models that follow the basic scheme of the original [HH](#) are called *HH-type models* and are probably the most successful ones in modeling membrane excitation.

2.3.4 Cable Equation

It is important to note that the HH model in section 2.3.3.2 only represents a single patch of axon, as there is no spatial dependence. This corresponds to reducing the neuron to a point in space. In the real world, however, neurons can show quite complex morphologies, and channel types and densities can vary significantly across different parts of the neuron. In search of a mathematical model for this, neuroscientists have rediscovered cable theory – originally developed to study signal decay in underwater telegraphic cables by Lord Kelvin – to describe the potential spread in complicated neuronal morphologies (cf. e.g. [84]). The *cable equation* reads

$$\frac{1}{R_a(x)} \frac{\partial^2 V}{\partial x^2} = C_m(x) \frac{\partial V}{\partial t} - I_{\text{memb}}(x) \quad (2.37)$$

with the axial (cytosol) resistivity R_a and the membrane capacitance C_m (see the schematic equivalent circuit in fig. 2.6). As in the case of the HH model, quantities may be expressed per unit length. In any case, $R_a(x)$ and $C_m(x)$ are depending on the position-dependent radius $r(x)$ of the neurite.

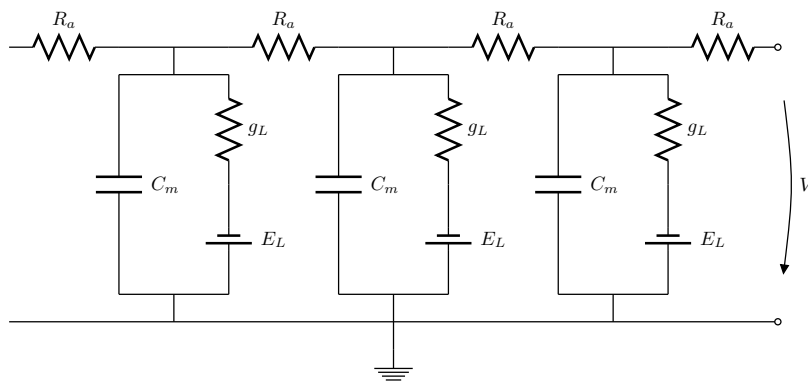


Figure 2.6: **Equivalent circuit for a spatially extended neuronal membrane.** Multiple (in this case passive) membrane patches are connected through axial resistances R_a , which in general are position-dependent.

This equation is a one-dimensional parabolic diffusion-reaction PDE and essentially describes the potential propagation as a transient diffusive process along a cylinder of varying radii, where the signal velocity depends on cytosol resistivity R_a (and implicitly also the fiber diameter $d = 2r(x)$). The right-hand side

corresponds to the HH eq. (2.30), where all trans-membrane contributions to the intracellular potential are lumped together into a single term $I_{\text{memb}}(x)$.

The cable equation has been very successful for modeling excitation of spatially extended neurons, and the simulation program NEURON [53] is the de-facto standard among neuroscientists.

2.3.5 Extracellular Space

So far we have only seen models of (parts of) the neuronal membrane and the signal propagation *within* neurons. What is missing in these models is the extracellular space. This is an important structure, as large amount of experimental data today is obtained by extracellular measurements. From the theoretical point of view, it is therefore desirable to have a suitable model of how signals are transmitted in extracellular space.

One more note on the wording. In the literature, the term **LFP** is used with slightly different meanings, apart from the fact that it is poorly named anyway. In the most general sense, it describes the potential time course obtained at a single point in extracellular space as a superposition of potential contributions stemming from a number of surrounding neurons. In most experimental contexts, however, the term describes an already low-pass filtered signal, which does not contain any fast components like those from action potentials. For the latter, the term **extracellular action potential (EAP)** has found its way into the terminology. It should be noted that in the following, when the term **LFP** is used, we denote with this the *unfiltered* potential at any point in space, regardless of the number of contributing cells (in the present case, only one or two). Since we only consider axonal membrane currents following an **AP** as contributions, it should be clear that this is essentially synonymous to an **EAP** in the absence of any synaptic currents or contributions by other cells, in contrast to the common usage.

2.3.5.1 Volume Conductor Theory

The field of volume conductor theory deals with the propagation of electric and magnetic fields within volume conductors like the brain [81]. It therefore provides a model of the extracellular space, without explicitly representing its complex geometry. Instead, it is characterized as a homogeneous medium with an effective conductivity σ . It is based on the quasi-static approximation of Maxwell's equations by neglecting the influence of magnetic fields on the electric field and the existence of any free charges in the medium. The electromagnetic part can be neglected by estimating its effect in the relevant frequency range and

finding that it is orders of magnitude smaller than the electrostatic contributions. When the timescales for the balancing of free charge in the medium are compared with the timescales of neural excitation, the former is found to happen so fast it can be regarded as instantaneous, see [2] for an in-depth derivation. This results in a Laplace equation

$$\begin{aligned} \Delta\phi &= 0 && \text{in } \Omega \\ \phi &= g && \text{on } \Gamma_D \\ \sigma\nabla\phi \cdot \mathbf{n} &= j && \text{on } \Gamma_N \end{aligned} \quad (2.38)$$

for the potential due to *current densities* j at the cell boundary and a given conductivity σ of the extracellular medium. This equation is dual to the electrostatic equation

$$\begin{aligned} \Delta\phi &= 0 && \text{in } \Omega \\ \phi &= g && \text{on } \Gamma_D \\ \epsilon\nabla\phi \cdot \mathbf{n} &= f && \text{on } \Gamma_N \end{aligned} \quad (2.39)$$

due to a boundary charge density f and an electric permittivity ϵ of the medium. Note that both equations have a homogeneous right-hand side by the absence of any free charges in the medium.

Interestingly, eq. (2.38) uses current sources, which are inherently time-dependent. It therefore provides a stationary state for the extracellular potential which will establish given the conductivity of the medium. Following the above reasoning that this process happens instantaneously, the equation takes the same form as eq. (2.39), predicting an immediate potential response due to a given charge density.

2.3.5.2 Line Source Approximation

The line source approximation introduced in [57] is an analytical solution to the volume conductor eq. (2.38) under a certain geometry approximation, namely, the finite thickness of the neuron fibers is neglected and collapsed to a line. It is widely used as an effective model to compute the extracellular potential at any point in space, using only the values of the membrane currents at a finite number of line segments.

The equation for a point (r, h) due to a single line source j reads

$$\Phi_j(r, h) = \frac{\rho I_j}{4\pi\Delta s} \log \left| \frac{\sqrt{h^2 + r^2} - h}{\sqrt{l^2 + r^2} - l} \right|, \quad (2.40)$$

where I_j is the *total transmembrane current* of line j , r is the radial distance from the line of length Δs , h is the longitudinal distance from the end of the line, and $l = \Delta s + h$ is the distance from the start of the line. The parameter $\rho = \frac{1}{\sigma}$ describes the resistivity of the extracellular medium. The complete extracellular potential Φ at any point (r, h) of a neuron morphology consisting of a number M of connected lines is then simply given by the superposition of all line source potentials:

$$\Phi(r, h) = \sum_{j=1}^M \Phi_j(r, h). \quad (2.41)$$

This model is very convenient especially when using cable equation models based on a line segment approximation of the original neuron geometry, since all necessary data is readily available and the only free parameter is ρ . Note also that – in contrast to a numerical solution of eq. (2.38) – the LSA eq. (2.40) does not need to specify any boundary conditions next to the membrane currents, as it implicitly fulfills a potential of zero at infinity.

2.3.5.3 The Poisson-Nernst-Planck Equations of Electrodifffusion

The most general approach to model a neuron and the surrounding space is to consider the – in the true sense of the word – atomic computation unit of the brain: the ion. In contrast to the approach of volume conductor theory in section 2.3.5.1, free charges are not neglected, but modeled explicitly through ion concentrations.

A suitable model for the movement of ions in a static solvent considers both chemical diffusion (due to a concentration gradient) and electrostatic drift (attraction/repulsion due to a potential gradient), often summarized by the term *electrodifffusion*.

This is described by the Nernst-Planck equation

$$\frac{\partial n_i}{\partial t} + \nabla \cdot \mathbf{F}_i = 0 \quad (2.42a)$$

with the ion flux

$$\mathbf{F}_i = -D_i (\nabla n_i + z_i n_i \nabla \phi) , \quad (2.42b)$$

where n_i , $i = 1, \dots, N$ are defined as relative concentrations (with respect to a scaling concentration $n^* = N_A$, the Avogadro constant) with units mM for the N different ion species, z_i (as before) is the valence and D_i the (position-dependent) diffusion coefficient of ion species i . Together with the Poisson equation for the electric potential

$$\nabla \cdot (\epsilon \nabla \phi) = -\frac{e^2 n^*}{\epsilon_0 k_B T} \sum_i z_i n_i , \quad (2.43)$$

this constitutes the **PNP** system.

Equation (2.42) is of convection-diffusion type and describes the time-dependent change in concentrations due to diffusion and drift through an electrical field, while the elliptic eq. (2.43) gives the electric potential ϕ at any point in space. We can see that each **PDE** is linear in its primary variable, but together the system is nonlinear due to the coupling.

Neglecting the finite ion size and representing the ion concentrations as continuous variables in a mean-field approach was found to be valid [67], as the typical size of an ion (about $1 \text{ \AA} = 100 \text{ pm}$, the extent of the *Stern layer*) is one order of magnitude smaller than the smallest relevant spatial scales (Debye length, about 1 nm for typical ion concentrations in body fluids).

Let us finally spend a few words on the analysis of solutions for the general 3D **PNP** system. The majority of publications using the **PNP** system come from the field of semiconductor analysis, although many of them also deal with the electric fields inside ion channels and around biomolecules. [69] provides a good starting point for this, as it gives a comprehensive overview over previous works involving the **PNP** equations and also touches the topic of its mathematical analysis. Existence and stability of solutions has been shown for the steady-state case [60]; solutions are presumed not to be unique and it is assumed that restrictions on the boundary conditions would have to be made for this to hold.

For the instationary case under closed-system boundary conditions, existence and convergence to stationary solutions for long-time behavior could be shown [22]. In [38], the system is analyzed in the presence of permanent charges, explicitly mentioning ion channels as one source. In [105], existence and uniqueness of global weak solutions to the general drift-diffusion-Poisson system are shown in

the presence of an additional reaction term; references therein treat this system under various boundary conditions.

The literature is quite extensive, but we are not aware of a statement on the well-posedness of the general problem eqs. (2.42) and (2.43) under arbitrary boundary conditions, especially not for the model at hand – which will be introduced in the following chapter – using two (intra- and extracellular) electrolytes separated by a membrane with imposed nonlinear boundary flux conditions by the HH ion channel model.

After this tour through different neuron models, we have now arrived at the most general continuum model. On first sight, this system only describes ion movements in electrolytes. We will see in the following chapter how to include the membrane and its ion channels into this framework in order to obtain a very detailed electrodiffusion model of a neuron and its surrounding.

3 General Model of an Axon in Extracellular Space

Numerik ist ein sehr schmutziges Geschäft.

(Hans-Peter Gail)

The model considered throughout this work will deal exclusively with the axonal part of the neuron. We first mention the governing equations and their boundary conditions, then the numerical methods used to solve those equations in an efficient and accurate way are presented. This chapter is largely based on an edited version of [82].

3.1 General Remarks

In section 1.3, we mentioned the basic assumptions most cable equations models are based on today. Half of those can be dropped when using the PNP equations of electrodiffusion.

We do not neglect changes in the radial direction (1), nor do we regard the extracellular space as isopotential (2). Most importantly, we explicitly represent the concentrations as primary variables, which are allowed to change in space and time, in contrast to assumption 6. Furthermore, we couple intra- and extracellular space, such that the assumption of independence (7) can be discarded as well.

The remaining assumptions are supposed to hold, i.e. the validity of the mean-field approach that allows us to represent all quantities as continuous variables (4), neglecting magnetic fields (3) and the equivalent circuit assumption from the HH model (8). Assumption 5, which states that the ES can be regarded as a homogeneous medium, is not a necessary one in our model, as permittivity ϵ and diffusion coefficients D_i may in principle be position-dependent. However, we chose these coefficients to be homogeneous and constant in all considered cases.

3.2 A Simplified Axon Model in Cylindrical Coordinates

As the solution of a full 3D system is computationally expensive, we exploit the rotational symmetry of an idealized unbranched axon. By representing the computational domain in cylinder coordinates and assuming there is no change in angular direction, it can be reduced to two dimensions. This enables the calculation of 3D results with a drastically reduced computational complexity. For the numerical solution, the rectangular elements of the computational grid will be treated as (hollow) cylinders, and the volumes and integration elements will be calculated accordingly.

In the upper part of fig. 3.1, the cylinder geometry is shown with the two-dimensional subset highlighted, constituting the effective computational domain. The x -axis represents the domain's symmetry axis in *axial* direction (usually denoted h or z in cylinder coordinates), eliminating the angular coordinate θ from the equations. The y -axis is usually denoted by r or ρ in cylinder coordinates and stands for the *radial* direction. For historical reasons¹ we will stick to the variables x and y for the coordinate axes in the following.

The domain consists of three partitions: cytosol, membrane, and extracellular space. Cytosol and extracellular space are electrolytes, yielding the electrolyte domain $\Omega_{\text{elec}} = \Omega_{\text{CY}} \cup \Omega_{\text{ES}}$. It may contain an arbitrary number N of concentrations of different ion species. Here, however, we will restrict ourselves to the minimal set of $N = 3$ monovalent species sodium (Na^+), potassium (K^+), and chloride (Cl^-). Sodium and potassium are needed for the ion channel dynamics triggering an action potential; chloride is a representative of the anions needed for electroneutrality in the bulk solution and does not cross the membrane in this model.

The membrane domain Ω_{memb} separates the two parts of the electrolyte domain, therefore Ω_{elec} is not connected. Ω_{elec} and Ω_{memb} form a partition of the computational domain: $\bar{\Omega} = \bar{\Omega}_{\text{elec}} \cup \bar{\Omega}_{\text{memb}}$.

3.3 PNP Equations and Boundary Conditions

This setup necessitates the introduction of additional boundary conditions on the membrane-electrolyte-interface Γ_{int} – next to the obligatory boundary conditions on the domain boundary Γ_{ext} – such that the set of all boundary points is given

¹Previous versions of this model were in 1D and plain 2D, with the coordinate axes named accordingly.

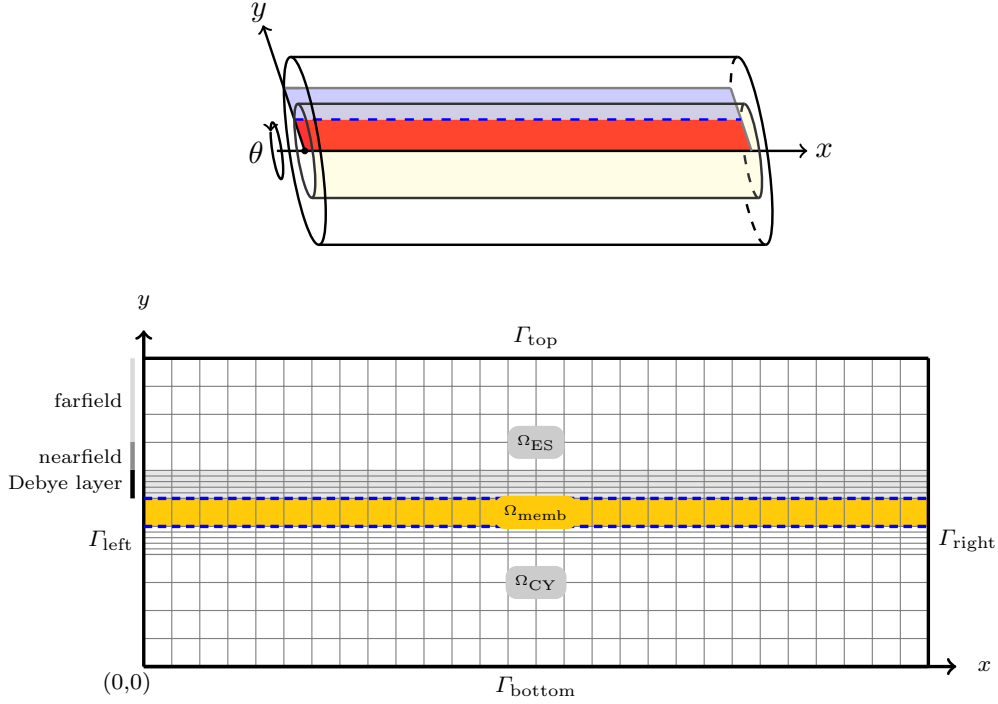


Figure 3.1: **Two-dimensional computational domain for the cylinder-symmetric axon model.** The upper part shows the cylinder geometry into which the 2D computational grid is embedded, assuming continuous symmetry in the angular direction. In the lower part, the domain boundary Γ_{ext} is represented by solid lines, while the interior (electrolyte-membrane) boundary Γ_{int} is plotted with dashed lines. This divides the domain into three subdomains: the (non-connected) electrolyte domain $\Omega_{\text{elec}} = \Omega_{\text{CY}} \cup \Omega_{\text{ES}}$ consisting of two subdomains and the separating membrane subdomain Ω_{memb} . The Debye layer of Ω_{ES} close to the membrane is highlighted in gray, followed by the nearfield and farfield parts. The lower boundary represents the inner-cell symmetry axis. Note that this scheme is not to scale, as the actual mesh sizes in y -direction differ by several orders of magnitude between Debye layer and farfield, making the grid very anisotropic.

by $\Gamma = \Gamma_{\text{int}} \cup \Gamma_{\text{ext}}$. To properly define the boundary conditions in this setup in a general way, we first need to introduce a little bit of notation for the description of the boundaries and their corresponding boundary condition types.

As can be seen in fig. 3.1, the exterior boundary $\Gamma_{\text{ext}} = \Gamma_{\text{bottom}} \cup \Gamma_{\text{left}} \cup \Gamma_{\text{right}} \cup \Gamma_{\text{top}}$ consists of four parts, and the interior boundary $\Gamma_{\text{int}} = \Gamma_{\text{CY}} \cup \Gamma_{\text{ES}}$ of two non-connected parts at either side of the membrane.

To increase the notational complexity even more, each of these boundary subsets may again be a partition of two subsets for Dirichlet and Neumann conditions, denoted schematically by subscripts $\Gamma_{\cdot,D}$ and $\Gamma_{\cdot,N}$, respectively. Additionally, the boundary condition type may be different for each equation of the PNP system. In this context, however, it will not be necessary to distinguish between individual concentrations; it is sufficient to have one set of boundaries for each equation type, denoted by the addition of superscripts $\Gamma_{\cdot}^{\text{P}}$ and $\Gamma_{\cdot}^{\text{NP}}$ for Poisson and Nernst-Planck equation, respectively.

In summary, we arrive at the final notation of a boundary in the schematic form $\Gamma_{L,T}^{(E)}$, where L specifies the *location* of the boundary, T the boundary condition *type* (Dirichlet or Neumann), and E the *equation* for which it is defined (Poisson or Nernst-Planck).

With this notation of the boundaries, we now have the Nernst-Planck eq. (2.42) defined on Ω_{elec} with boundary conditions

$$\mathbf{F}_i \cdot \mathbf{n} = j_i^{(\text{NP})} \quad \text{on } \Gamma_{\text{ext},N}^{(\text{NP})} \cup \Gamma_{\text{int}}^{(\text{NP})} \quad (3.1a)$$

$$n_i = g_i^{(\text{NP})} \quad \text{on } \Gamma_{\text{ext},D}^{(\text{NP})} \quad (3.1b)$$

and the Poisson eq. (2.43) defined on the whole domain Ω with boundary conditions

$$\epsilon \nabla \phi \cdot \mathbf{n} = j^{(\text{P})} \quad \text{on } \Gamma_{\text{ext},N}^{(\text{P})} \quad (3.2a)$$

$$\phi = g^{(\text{P})} \quad \text{on } \Gamma_{\text{ext},D}^{(\text{P})} , \quad (3.2b)$$

where \mathbf{n} denotes the unit outer normal vector. For consistency, of course, $\Gamma_{\text{ext},N}^{(\cdot)} \cup \Gamma_{\text{ext},D}^{(\cdot)} = \Gamma_{\text{ext}}$ holds for each equation. From eq. (3.1) we see that the internal concentration boundary is always a Neumann boundary, as we want to describe membrane *fluxes*, which are equivalent to the more commonly used term of membrane *currents* by a constant factor ezn^* . For the external boundaries, any combination of boundary conditions is possible in principle. Nevertheless, as the solution of the Poisson eq. (2.43) is only defined up to a constant, we need at

least one point with a Dirichlet condition in order to have a well-posed problem with a unique solution.

One more word about the domains of eq. (2.42) and eq. (2.43), which are obviously different: the Poisson equation is defined (and continuous) on the whole domain, while the Nernst-Planck equation is only defined on the electrolyte subdomain. This means that, for simplicity, we assume the membrane to be free of charge carriers. (Fixed) membrane surface charges could easily be added as an additional source term f_P in the Poisson equation, but those are not considered here.

For the Nernst-Planck equation, the precise locations of eqs. (3.1) and (3.2) may be problem-dependent, but in most considered cases they will be the same. Therefore, we give the definition that holds for the majority of setups hereafter; if boundary conditions are defined differently, this will be explicitly stated in the following. The general scheme reads

$$\begin{aligned} j_i^{(\text{NP})} &= \begin{cases} f_i^{\text{memb}}(n_i, \phi, t) & \text{on } \Gamma_{\text{int}}^{(\text{NP})} \\ 0 & \text{on } \Gamma_{\text{ext},N}^{(\text{NP})} = \Gamma_{\text{ext}} \setminus \Gamma_{\text{top}} \end{cases} \\ g_i^{(\text{NP})} &= n_i^0 & \text{on } \Gamma_{\text{ext},D}^{(\text{NP})} = \Gamma_{\text{top}} \\ j_i^{(\text{P})} &= 0 & \text{on } \Gamma_{\text{ext},N}^{(\text{P})} = \Gamma_{\text{ext}} \setminus \Gamma_{\text{top}} \\ g_i^{(\text{P})} &= 0 & \text{on } \Gamma_{\text{ext},D}^{(\text{P})} = \Gamma_{\text{top}} , \end{aligned}$$

where we call the constant n_i^0 the *bulk concentration* of species i . We can see that in this general scheme the locations of Neumann and Dirichlet boundary conditions match for Poisson and Nernst-Planck equations, respectively.

The membrane flux function f_i^{memb} is the heart of this model and will be defined in the next section. The Dirichlet boundary conditions for the concentrations $g_i^{(\text{NP})}$ on the upper exterior boundary model an infinite reservoir for each ion species. The potential is clamped to zero at the upper extracellular boundary by means of $g^{(\text{P})}$, which introduces an error equal to the value of the real potential value $\hat{\phi}(y_{\text{max}})$ calculated for an unrestricted domain, where the potential is 0 for $y \rightarrow \infty$.

For a point charge on the membrane, this error would correspond to an absolute shift at each point in the domain. As the potential of a point charge falls off as $\frac{1}{r}$ in 3D, an increase of the domain size by a factor 100 will reduce the error by a factor $\frac{1}{100}$. For a finite line charge (like in the present case of an active membrane), this is only true if the radial distance is large compared to the length of the line charge. Otherwise, the potential drop is a logarithmic function of the

membrane distance (cf. LSA, e.g. in [45] and eq. (2.40)). In any case, increasing the domain size in y -direction will reduce the error introduced by the upper Dirichlet boundary. A sufficiently large domain size of $y_{\max} = 10$ mm was chosen to account for this.

At the lower boundary, which represents the intracellular symmetry axis, the potential gradient and ion fluxes are vanishing by the definition of $j_i^{(\text{NP})}$ and $j^{(\text{P})}$, such that no boundary artifacts are introduced.

3.4 Derivation of the Membrane Flux

The most important part for the boundary conditions – and, as we will see later on, the dynamics of the system as a whole – are the membrane currents. To represent those, we use the HH system from section 2.3.3.2 in a slightly modified form: the leak channel was split into two separate leak channels for Na^+ and K^+ , respectively. The corresponding conductances now read

$$g_{\text{Na}} = g_{\text{Na}_v} + g_{\text{Na}_L} \quad (3.3)$$

$$g_{\text{K}} = g_{\text{K}_v} + g_{\text{K}_L} \quad (3.4)$$

$$g_{\text{Cl}} = 0, \quad (3.5)$$

where the voltage-dependent parts g_{Na_v} and g_{K_v} of the total conductances g_{Na} and g_{K} are given in eq. (2.34a) as before. The leak parts g_{Na_L} and g_{K_L} add up to the total leak conductance of the original HH model,

$$g_{\text{Na}_L} + g_{\text{K}_L} = g_L.$$

The reason for splitting up the leak channel becomes apparent when we look at the Goldman eq. (2.28): here, different permeabilities of the membrane for different ion species determine the membrane potential, given the intra- and extracellular bulk concentrations. When we now think in terms of HH-type ion channels and, hence, in terms of conductances instead of permeabilities, the PCM eq. (2.29) is the right choice for the membrane potential, using *relative leak conductances* $\frac{g_{\text{Na}_L}}{g_L}$ and $\frac{g_{\text{K}_L}}{g_L}$.

Whether we take the view of permeabilities (Goldman) or conductances (PCM), it is the *relative* quantities of channels that matter. We need a means to express these relative values in the model, since a single channel – as in the original HH model – is not sufficient if we include concentrations explicitly.

For the definition of the membrane flux, each point $\mathbf{x} \in \Gamma_{\text{CY}}$ on the cytosol-

membrane interface is associated with a point $\mu(\mathbf{x}) \in \Gamma_{\text{ES}}$ on the opposite membrane-extracellular interface by a map $\mu(\mathbf{x}) = \mathbf{x} + d_{\text{memb}} \cdot \mathbf{n}_{\text{CY}}$, where d_{memb} is the membrane thickness and \mathbf{n}_{CY} is the unit outer normal at \mathbf{x} pointing in the direction from cytosol to membrane. The values of potential and concentrations evaluated at these points are called $\phi^{\text{CY}} = \phi(\mathbf{x})$, $n_i^{\text{CY}} = n_i(\mathbf{x})$, $\phi^{\text{ES}} = \phi(\mu(\mathbf{x}))$ and $n_i^{\text{ES}} = n_i(\mu(\mathbf{x}))$.

We now take the single Hodgkin-Huxley membrane current

$$I_i^{\text{memb}} = g_i(\phi, t)([\phi] - E)$$

for the membrane potential $[\phi] = \phi^{\text{CY}} - \phi^{\text{ES}}$ and replace the constant battery E by a variable concentration-dependent reversal potential calculated from the Nernst eq. (2.27). After adding the necessary scaling factors to bring $f_i^{\text{memb}} = I_i^{\text{memb}}/ezn^*$ to SI units mol/(m² s), we obtain the concentration-, potential- and time-dependent *membrane flux* f_i^{memb} of species $i \in \{\text{Na}, \text{K}\}$,

$$f_i^{\text{memb}}(\mathbf{x}) = f_i^{\text{memb}}(\mu(\mathbf{x})) = g_i(\phi, t) \frac{kT}{e^2 z^2 n^*} \left(z[\phi] + \ln \frac{n_i^{\text{ES}}}{n_i^{\text{CY}}} \right). \quad (3.6)$$

Note that two opposite points on the membrane interface are identified with each other here, i.e. the membrane thickness is essentially neglected. This is a compulsory assumption with respect to the underlying membrane model, which was replaced by an equivalent circuit by the HH model in fig. 2.5, where the spatial extent of the membrane is inherently not represented.

But this is also reasonable from the physical point of view, as the membrane thickness (in the range of a few nm) is so small that any delay from an ion crossing the membrane can be neglected and considered instantaneous in view of the governing AP time scales in the range of milliseconds. Of course, the Poisson equation is not affected by this. The correct potential fall-off over the membrane is respected, as its spatial extent is represented explicitly.

This interior boundary condition fits nicely into our framework, as it unifies the potential-dependent HH system with the concentration-dependent Nernst equation to arrive at an expression that represents all the features of the potential- and concentration-dependent ion flux \mathbf{F}_i of the PNP system. An equilibrium state of this model is expected to satisfy the Nernst eq. (2.27) (for a single ion species) or the PCM eq. (2.29) (for multiple species) for the potential as well as the concentration distribution eq. (2.23) predicted by the Poisson-Boltzmann eq. (2.22). This will be tested in chapter 5.

3.5 Numerical Methods

3.5.1 Weak Form and Discretization

For the numerical solution of the PNP system as defined in section 2.3.5.3 and with boundary conditions given in section 3.3, we use the finite element method introduced in section 2.2 with piecewise linear, globally continuous $Q_1(\mathcal{T})$ nodal basis functions on an axi-parallel rectangular grid, as defined in section 2.2.2.5.

More specifically, the computational grid \mathcal{T} will always be *tensor grid*, i.e. it can be written as the Cartesian product of the two coordinate vectors, $\mathcal{T} = X \times Y$. This allows for several optimizations in the numerical code, which turn out to be beneficial for the computational efficiency.

For application of the FEM, we obtain the weak formulation in residual formulation for each equation i , $i = 1, \dots, N + 1$ of the PNP system by multiplication with test functions v_i , integrating over the respective domain and applying integration by parts. We assume the v_i have been chosen to fulfill the respective Dirichlet boundary conditions $g_i^{(\text{NP})}$ and $g^{(\text{P})}$.

For the Nernst-Planck equation, the temporal part

$$R_{\text{NP},\text{T}} = \int_{\Omega_{\text{elec}}} \frac{\partial n_i}{\partial t} v_i dx, \quad i = 1, \dots, N \quad (3.7a)$$

and the spatial part

$$\begin{aligned} R_{\text{NP},\text{S}} &= \int_{\Omega_{\text{elec}}} D_i (\nabla n_i + z_i n_i \nabla \phi) \cdot \nabla v_i dx \\ &+ \int_{\Gamma_{\text{N}}} j_{\text{NP}} v_i ds, \quad i = 1, \dots, N \end{aligned} \quad (3.7b)$$

are combined yielding

$$R_{\text{NP}} = R_{\text{NP},\text{T}} + R_{\text{NP},\text{S}}. \quad (3.7c)$$

For the Poisson equation, we get

$$\begin{aligned} R_{\text{P}} &= \int_{\Omega} -\epsilon \nabla \phi \cdot \nabla v_{N+1} + \left(\frac{e^2 n^*}{\epsilon_0 k T} \sum_i z_i n_i \right) v_{N+1} dx \\ &+ \int_{\Gamma_{\text{N}}} j_{\text{P}} v_{N+1} ds \end{aligned} \quad (3.8)$$

and therefore the residual for the full system reads

$$R = \begin{pmatrix} R_{\text{NP}} \\ R_{\text{P}} \end{pmatrix}. \quad (3.9)$$

If we now denote the unknown function by $u = (n_1, \dots, n_N, \phi)^T$ and the vector of test functions by $v = (v_1, \dots, v_{N+1})^T$, the finite element problem according to section 2.2.2.4 can be written as

$$\text{Find } u \in Q_1^{N+1}(\mathcal{T}) : \quad R = R(u, v) = 0 \quad \forall v \in Q_1^{N+1}(\mathcal{T}). \quad (3.10)$$

By applying the method of lines, each equation is discretized in space first by representing the unknown functions n_i and ϕ as well as the test functions v_i by $Q_1(\mathcal{T})$ nodal basis functions on the tensor grid \mathcal{T} , and then in time using the implicit Euler time-stepping scheme from eq. (2.21).

3.5.1.1 Space Discretization

As suggested in fig. 3.1, the grid is refined toward the membrane in y -direction. This is essential in order to resolve the Debye length, the characteristic length scale over which the electrolyte ion concentrations deviate significantly from their bulk values close to the membrane (see section 2.3.2.2).

In x -direction, the grid is allowed to be much coarser, as there is no Debye layer to resolve. This results in a very *anisotropic* grid, especially at grid cells close to the membrane with ratios of up to $\frac{h_x}{h_y} = 200\,000$ between mesh sizes in x - and y -direction.

The cylinder symmetry introduces another subtle difficulty for the numerical treatment of the system. Since the cell volumes increase super-linearly in positive y -direction (roughly with $y \, dy$), the entries of the full residual R in eq. (3.9) differ by several orders of magnitude (10^9 for a domain size of 10 mm) solely by the presence of volume integrals in the weak form of the equations. This imposes a severe difficulty for the linear solver.

A *threshold volume scaling* strategy is applied to account for this: at a certain distance from the membrane, a reference volume V_{ref} is calculated. All entries of the residual from an unknown at node i , where the corresponding volume V_i is greater than V_{ref} , are scaled by a factor V_{ref}/V_i . Here, V_i is defined as the minimum volume of all adjacent cells of node i . So we are essentially compensating for the large cell volumes by scaling down those residual entries that stem from cells with a volume larger than the threshold volume V_{ref} . By choosing a certain threshold volume V_{ref} , we can exclude the cytosol, membrane and Debye layer cells close

to the membrane from this scaling procedure. These cells are comparably small in volume, but contribute large entries to the residual because of the exhibited gradients close to the membrane. Therefore, scaling these residual entries would increase those already large entries, which is not desired. The threshold volume is chosen such that only residual contributions from large volume elements in more distant extracellular space are scaled down.

Mathematically, this corresponds to multiplying a diagonal matrix from the left to the linear system, meaning that the *same* linear system is solved in each Newton iteration. This scaling greatly improves the convergence properties of the Newton algorithm for this cylinder geometry setup.

3.5.1.2 Time Discretization

The choice of the time-stepping scheme strongly depends on the Péclet number (cf. eq. (2.6)), or, in the discrete case, the maximum *cell Péclet number* of the Nernst-Planck eq. (2.42),

$$\text{Pe}_{h,\max} = \max_{\substack{t_k \in \mathcal{T} \\ i \in 1, \dots, N}} \left\| \frac{\mathbf{h}_k \cdot \boldsymbol{\omega}_i}{D_i} \right\|_{\infty}$$

with mesh size vector $\mathbf{h}_k = \begin{pmatrix} h_x \\ h_y \end{pmatrix}$ and velocity vector $\boldsymbol{\omega}_i = D_i z_i \nabla \phi$ evaluated on each cell t_k , respectively.

A value of $\text{Pe}_h > 1$ means the advective (hyperbolic) part of eq. (2.42) is dominating, otherwise the equation is diffusion-dominated and it is more parabolic. Usually, for $\text{Pe}_h > 2$, an explicit time-stepping scheme is used to account for the advection-dominance limiting the maximally possible time step size. In the remaining cases, an implicit scheme will yield a stable solution and enable the use of larger time steps due to the diffusion dominance.

We see that the scalar diffusion coefficient D_i cancels out in the above equation, such that, for monovalent ions, the condition $\text{Pe}_{h,\max} < 1$ is satisfied if the potential does not vary by more than 1 (corresponding to about 25 mV) over the extent of one grid cell.

For the present model with parameters in the physiological range and a mesh resolving the Debye layer, the potential gradient is always sufficiently fine resolved. Thus, the maximum grid Péclet number was always significantly smaller than 1, and the implicit Euler scheme was chosen. Since the system is very susceptible to numerical oscillations, the choice of an implicit scheme provides an additional benefit. Implicit schemes tend to smooth out these unphysical oscillations over time, while they may be amplified in explicit schemes.

The HH system as the driving force of an AP shows a great variability of the time scales on which the system dynamics act. During an AP, membrane fluxes and potential differences are large and change rapidly, so a small time step is needed to capture the dynamics during this period. On the other hand, potential differences during inter-spike intervals are small and so are the magnitudes of ion fluxes, allowing for the use of a larger time step. Therefore, an adaptive time-stepping strategy is used to speed up the simulation by controlling the time step Δt depending on the dynamics of the system.

The time step is bounded by minimum and maximum values of $\Delta t_{\min} = 0.05 \mu\text{s}$ and Δt_{\max} , respectively. The value of Δt_{\max} depends on the problem and will range between $10 \mu\text{s}$ – $50 \mu\text{s}$. During an action potential (membrane potential $[\phi] > -50 \text{ mV}$) or when an external stimulation is present, the maximum time step is additionally limited to $\Delta t_{\max, \text{AP}} = 10 \mu\text{s}$.

Apart from these fixed bounds, the change of the time step depends on the number of Newton iterations it_k needed to complete the previous time step k :

$$\Delta t_{k+1} = \begin{cases} \Delta t_k \times 1.1 & it_k < it_{\max} \wedge it_k \leq it_{k-1} \\ \Delta t_k / 1.2 & it_k > it_{\min} \\ \Delta t_k & \text{else .} \end{cases} \quad (3.11)$$

The upper and lower iteration bounds for adjusting the time step, it_{\max} and it_{\min} , depend on the problem.

3.5.2 Solving the PDE System

As seen in section 2.3.5.3, each equation eq. (2.42) and eq. (2.43) alone is linear in its unknowns. One could therefore use an operator-splitting approach and alternately solve the equations by a linear solver until convergence, in each time step. However, it could be observed that a very small time step (in the order of nanoseconds) is necessary to solve the system this way.

Since the nonlinearity of the whole system results from the coupling of both equations, it seems reasonable to represent this crucial feature also in the numerical method. The system is therefore solved fully-coupled using Newton's method, which requires the solution of one single large linear system in each iteration.

As described in section 3.4, the dynamic channel conductances from the HH scheme are needed to calculate the total membrane flux f_i^{memb} . Hence, for each channel type i , one additional ODE per gating particle has to be solved in each time step. This is done using a simple implicit Euler step. The membrane flux is calculated *once* at the beginning of each time step and kept fixed for the whole

Newton iteration. This way, the unknowns from the HH scheme do not enter the full system matrix for the PNP system, which avoids convergence issues arising from changing boundary conditions. However, this introduces a small splitting error of the order of $\mathcal{O}(\Delta t)$.

The time step size is adapted in a rather conservative manner (cf. section 3.5.1.2). Still, the Newton iteration might not converge for a chosen time step value Δt for various reasons. A restart mechanism was implemented to account for this: if the solver does not converge for a given time step Δt , the procedure is repeated for the halved time step $\frac{\Delta t}{2}$. A maximum number of three restarts is allowed, otherwise the simulation will be terminated.

A significantly higher stability was observed for the fully-coupled approach using Newton's method, leading to possible time steps of the order of tens of microseconds, as indicated by the values of time step thresholds from the previous section.

3.5.2.1 Linear Solvers

For small systems, SuperLU [33] was used, for larger systems with more than 50 000 unknowns, a **stabilized biconjugate gradient (BiCGStab)** iterative solver, preconditioned by an **inexact LU (ILU)** decomposition, turned out to be faster while maintaining the same accuracy. A combination of a restarted **generalized minimal residual (GMRes)** method in combination with an **algebraic multigrid (AMG)** preconditioner proved to be very robust and efficient when solving the system in parallel. Both **ILU** and **AMG** preconditioners are capable of coping with the grid anisotropy introduced by the spatial discretization in section 3.5.1.1, and the usage of state-of-the-art iterative linear solvers like **BiCGStab** and **GMRes** is crucial in order to attain a reasonable computation time for the problem at hand.

3.5.3 Validation of the Numerical Algorithm

To test the numerical algorithm from section 3.5 and to validate the implementation as a DUNE module, which will be covered in the next chapter, a number of simulations have been run and compared to recently found unsteady analytical solutions of the PNP equations for the case of a single electrolyte domain [89].

These solutions provide a useful test suite, as they allow for tests of the numerical methods with different boundary conditions and under different physical conditions. In particular, test cases for both diffusion- and advection-dominated conditions are presented.

For the rest of this section, we use a normalized version of the PNP system,

where the temperature T in eq. (2.43) is chosen such that the constant $-\frac{e^2 n^*}{\epsilon_0 k_B T}$ on the right-hand side of the Poisson equation is equal to 1. Similarly, the diffusion coefficient is fixed at $D = 1 \text{ m}^2/\text{s}$. As a consequence, the potential and concentrations are scale-free and it does not make much sense to express them in common physical units. Here, we consider them as dimensionless. The interval $[-5, 5]$ is used as the computational domain.

We now give a short overview over the three one-dimensional solutions² and provide numerical results that demonstrate the proper convergence behavior.

Closed system (*hamburger*) This solution describes an insulated system, i.e. it has zero boundary flux. The initial condition is smeared out over time, and, as such, this solution represents a diffusion-dominated case. We depict the solution for the potential and a single monovalent ion species (say, sodium) in fig. 3.2.

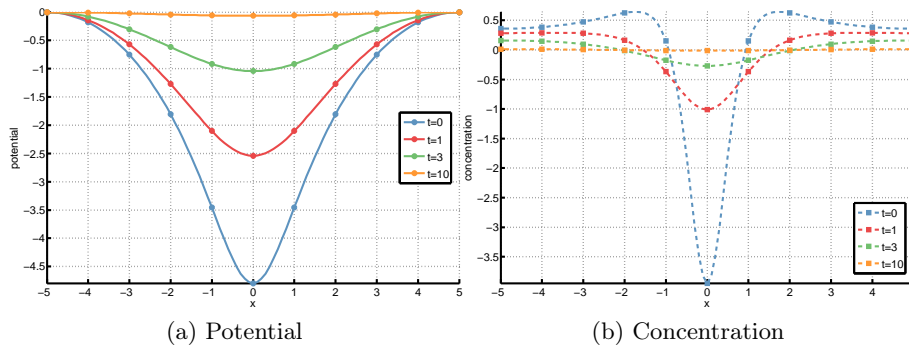


Figure 3.2: **Solution for hamburger example at different time points.**

In order to assess the convergence behavior in space and time separately, we first chose a very small time step $\Delta t = 10^{-9}$ and carried out 1000 time steps. At $t_{\text{end}} = 10^{-6}$, the errors in L^2 and L^∞ norms were calculated with respect to the analytical solution for different mesh sizes $h = 2^{-4}$ to 2^{-11} , which can be seen in fig. 3.3a. Due to the very small time steps, time discretization errors should be negligible compared to the spatial errors. This is confirmed when looking at the convergence order in fig. 3.3b, which shows the expected order of 2 for Q_1 finite elements as soon as the grid is fine enough to accurately resolve the initial condition.

²Since in 1D, the PNP can be transformed to the Burgers equation, each test case was named after a different burger, for no scientific reason.

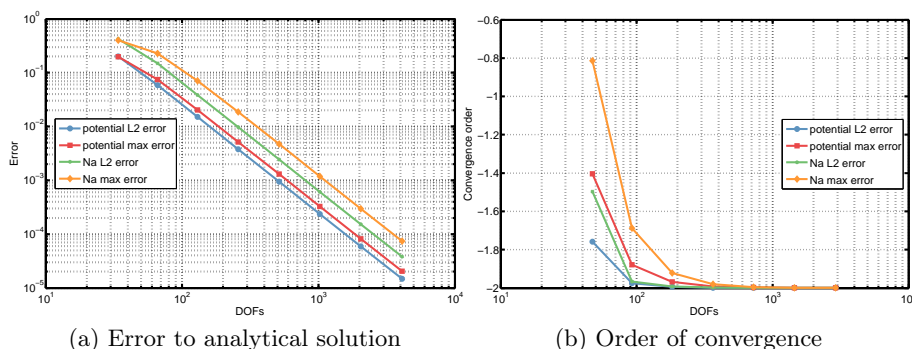


Figure 3.3: **Hamburger convergence behavior in space.**

Then, we chose a fine spatial resolution of 2^9 cells and assessed the temporal errors by varying the time step Δt from 1 down to 2^{-10} . At $t_{\text{end}} = 10$, errors were calculated as before, this time assuming spatial errors to be negligible, see fig. 3.4a. Again, we find the expected convergence order of 1 for the first order implicit Euler scheme in fig. 3.4b.

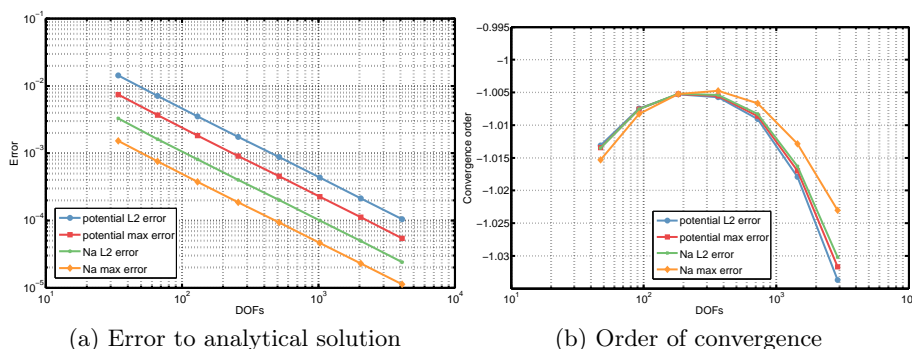


Figure 3.4: **Hamburger convergence behavior in time.**

Spatially homogeneous outflow (*cheeseburger*) The second example describes a system with a spatially uniform concentration and an outflow boundary condition. Over time, the concentration leaks out of the domain boundary due to a symmetric outward flow centered at $x = 0$. This results in a successive decrease of the spatially constant concentration, as visible in fig. 3.5. This example tests

the correctness of the numerical algorithm regarding non-zero time-dependent Neumann boundary conditions.

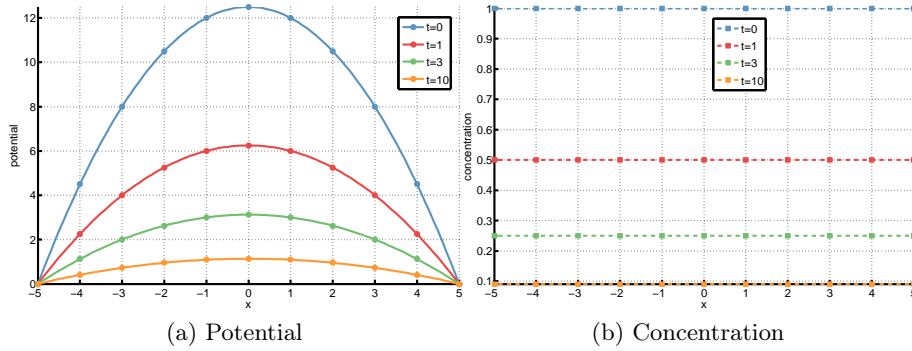


Figure 3.5: **Solution for cheeseburger example at different time points.**

Convergence in space for the potential is second-order, but this does not hold for the concentrations in fig. 3.7b. This result is intuitively clear respecting the concentration is spatially homogeneous. An increase in grid resolution can not be expected to yield an improvement for a function which can already be represented accurately with only a single cell. We confirm this by noting that the concentration error in fig. 3.6a is already close to machine precision at the coarsest resolution. Therefore, the strange behavior in the convergence order plot fig. 3.6b is uncovered to be numerical noise.

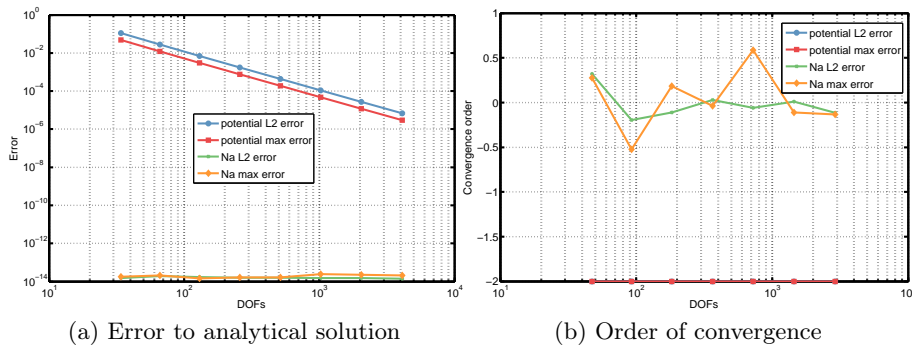


Figure 3.6: **Cheeseburger convergence behavior for spatial discretization.**

The behavior in time is not affected by this and shows first-order convergence

in fig. 3.7b as soon as the time step is sufficiently small. Interestingly, the concentrations initially show a slightly super-linear convergence speed.

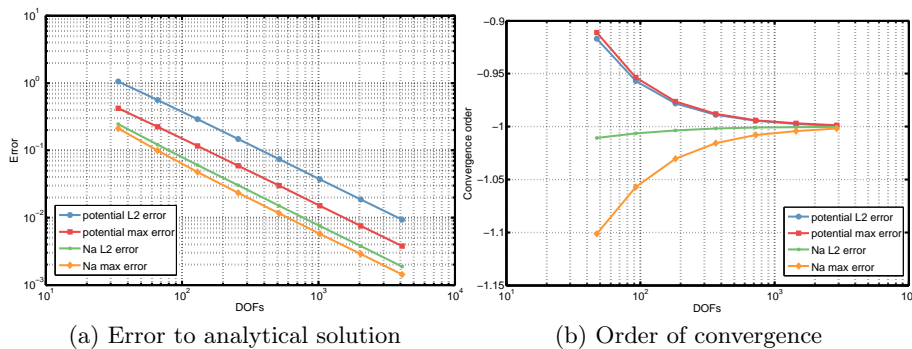


Figure 3.7: Cheeseburger convergence behavior in time.

Pure advection (*bicmac*) The last solution is purely advective, therefore the ion flux $\mathbf{F} = -D(\nabla n_i + z_i n_i \nabla \phi)$ is reduced to $\mathbf{F} = -D_i z_i \nabla \phi n_i$ and the drift velocity $v = D_i z_i \nabla \phi$ is replaced by a constant advection velocity $v = 1$. As a result, the initial concentration profile is simply transported to the right in fig. 3.8. This test case especially checks the numerical algorithm’s ability to cope with a drift-dominated system.

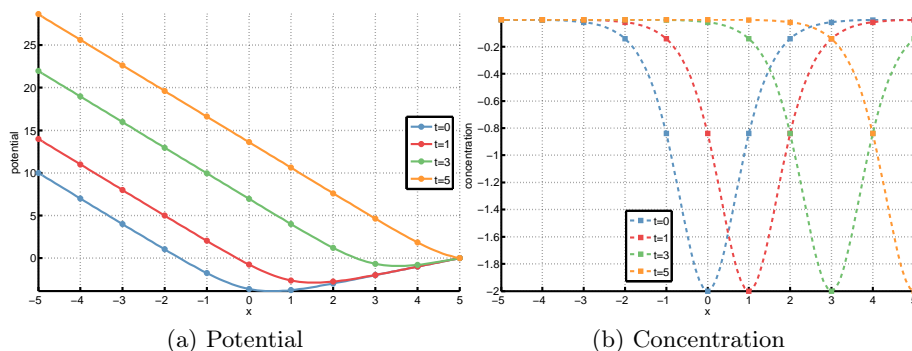


Figure 3.8: Solution for bigmac example at different time points.

This example is not as reluctant as the previous one and we see the full convergence order for both potential and concentrations, i.e. second order in space (fig. 3.9b) and first order in time (fig. 3.10b). In comparison to the

previous examples, the final convergence order is reached only for relatively fine discretization levels. The maximum concentration error is especially susceptible to this at coarser mesh sizes, since in these cases the transported concentration peak can not be captured exactly by the grid. We also see quite a delay in the time convergence at large time steps, which can be attributed to the smoothing character of the implicit Euler method.

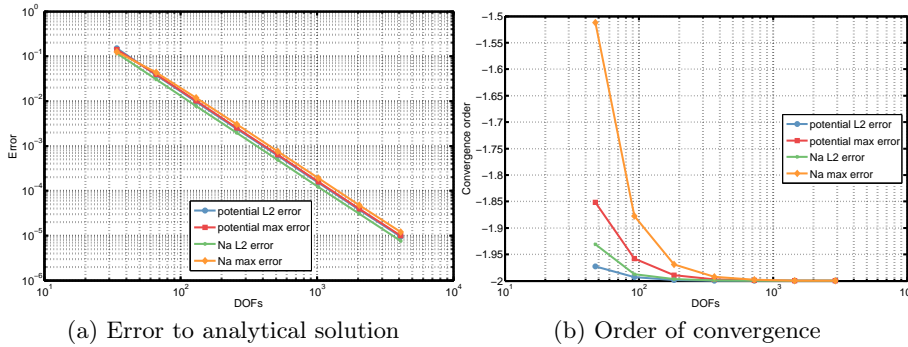


Figure 3.9: **Bigmac convergence behavior for spatial discretization.**

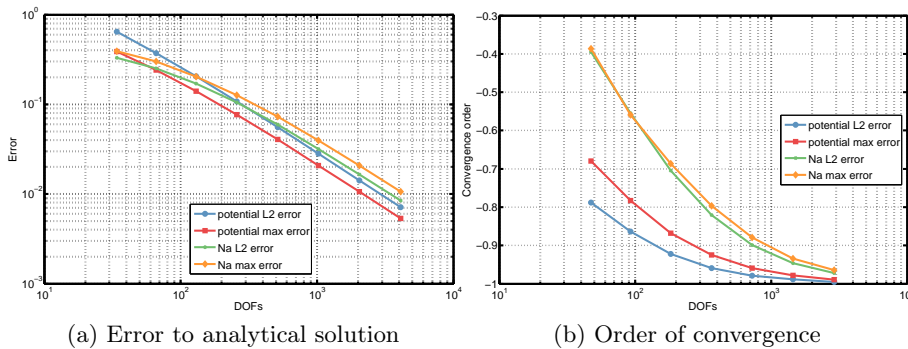


Figure 3.10: **Bigmac convergence behavior in time.**

Summarizing the validation by analytical solutions, we obtain the expected order of convergence in space and time for all test cases. Schönke [89] also describes a methodology to construct solutions in higher dimensions from the 1D solutions, but we refrain from carrying out extensive convergence analyses within the scope of this work.

4 Implementation

Warning: ISO C++ says that these are ambiguous, even though the worst conversion for the first is better than the worst conversion for the second.

(gcc 4.5)

This chapter strives to give a minimal overview over the implementation of the numerical algorithm from section 3.5.

4.1 The DUNE Framework

The implementation was done in C++ using DUNE (Distributed and Unified Numerics Environment) [13], a framework for the grid-based solution of PDEs. It consists of several modules, the heart of which is `dune-grid` containing the abstract grid interface [11] based on a rigorous mathematical description of hierarchical grids [12].

Existing grid implementations (or *grid managers*, in DUNE jargon), can be plugged into the framework by e.g. an adapter fulfilling the abstract interface. This is made possible by the extensive usage of C++ template techniques [102], which allows for the inclusion of arbitrary implementations without a big performance loss. This also summarizes the main design principles in DUNE: *flexibility* with regard to the reusability of software components and – at the same time – *efficiency* by removing the interface overhead at compile-time using generic programming techniques. The key here is to replace conventional inheritance (*dynamic polymorphism*) by *static polymorphism*, where the complete inheritance hierarchy is known at compile-time, thereby eliminating the runtime overhead of e.g. function table lookups necessary in dynamic polymorphism.

One main advantage of this approach is that one is not restricted to a certain grid implementation. There exist quite a number of grid managers with different features, but none of these is universal. Most grid managers are rather specialized towards a certain class of algorithms. DUNE makes it possible to simply exchange the underlying grid without having to rewrite the code for the numerical solution,

as it is based on the abstract interface. This has proven to be very handy in practice, as virtually any application can be solved within a single framework.

DUNE modules are classified into two groups: The *core modules* providing the basic functionality, and additional modules which extend the functionality of the core modules or implement a specific application. The core modules are

- **dune-common**: classes used by all DUNE modules, including data structures for dense vectors and matrices as well as the program `dunecontrol` providing the build system logic.
- **dune-geometry**: provides geometric information of the grid cells (included in the more general term of an *entity* in DUNE) based on generic *reference elements*, their mapping into the global space and quadrature rules for integration.
- **dune-grid**: contains the abstract grid interface and a small number of grid implementations as well as adapters for external grid managers.
- **dune-istl**: the [iterative solver template library \(ISTL\)](#) contains a number of iterative linear solvers and preconditioners that were designed specifically with respect to parallel efficiency.
- **dune-localfunctions**: this module defines functions living on the reference elements which can be used to assemble global finite element functions.

In addition, the following modules were used for the implementation:

- **dune-pdelab**: PDELab [15] is a discretization module which allows the user to specify a *local operator* living on a single grid cell. The functionality of PDELab then allows the generic assembly of the global matrix by means of a *grid operator* and its (sequential or parallel) solution by arbitrary combination of preconditioners and solvers from `dune-istl`. It also contains a Newton implementation.
- **dune-multidomaingrid**: this module provides a `Dune::MultidomainGrid` metagrid on top of a DUNE grid which allows for the definition of arbitrary subdomains, useful for multi-physics applications [77].
- **dune-multidomain**: this is an add-on module for PDELab which, in conjunction with `dune-multidomaingrid`, allows for the definition of different local operators on different subdomains of the `Dune::MultidomainGrid` [76]. The operators can be coupled in a very flexible way, integrating nicely into the PDELab framework.

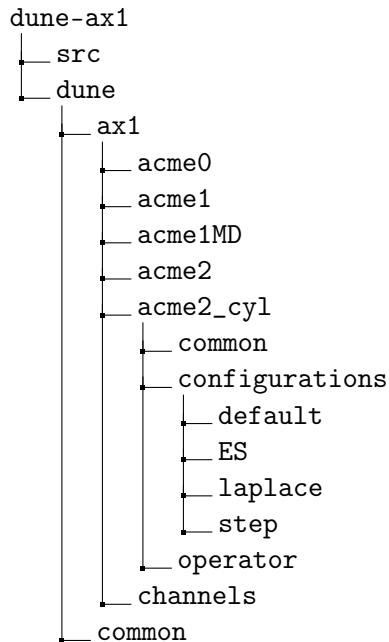
The joint functionality of these modules is used in the application module `dune-ax1`, whose main components will be described in the following section.

4.2 The DUNE Module `dune-ax1`

The module `dune-ax1` contains all the code that was used to implement the numerical solution of the model from section 3.2. The main actors in this module are listed below.

4.2.1 Directory Structure

The directory structure of the `dune-ax1` module looks as follows:



The `src` directory contains the application drivers, while the `dune/ax1` folder contains a number of subfolders which demonstrate the historical evolution of the application. The folders `acme0` to `acme2` represent the different evolution stages from a simple 1D model without a membrane on a `Dune::OneDGrid` to a 2D model with a membrane using `Dune::MultidomainGrid`. The reason for keeping the older stages was that initially the Subversion [96] version control system was used, which has its difficulties when using different development branches. After the switch to Git [44], keeping different versions in different branches has become quite easy, but the older stages are still there as a Subversion

legacy. The most recent version of the application resides in `acme2_cyl`. The folder `channels` contains classes representing a variety of HH-type ion channels that can be vectorized to a single `ChannelSet`. Additional classes used by every application can be found in `common`. The next section will describe the main classes in `acme2_cyl` and `common`.

4.2.2 Main Components

In the following, the main components responsible for the implementation are listed. Yet, this section does not only serve to list the purpose of each of these components; we will also highlight some of the concepts used, both with respect to programming techniques and to numerical subtleties.

`acme2_cyl_par.cc` This is the only `.cc` file in the application, all the others are header files containing (mostly templated) classes. This is due to the fact that, when using C++ templates, the usual separation between header and source files is no longer possible [61, p. 10]. The name contains the acronym “acme”, which stands for “active membrane”¹. The other building blocks of the name refer to the grid dimension (“2”), the cylindrical coordinate system (“cyl”), and the parallel solution of the equations (“par”).

`acme2_cyl_par.cc` as the application driver contains the `main()`-routine, which reads command line arguments and config file parameters, calls the grid generation procedure in `Ax1GridGenerator`, and sets up the `Dune::MultidomainGrid`. Afterwards, it calls `Acme2CylFactory`, which instantiates the central data class `Acme2CylPhysics` and hands all the objects over to `Acme2CylSetup`.

`Ax1GridGenerator` This class has a static method `generateTensorGrid()`, which fills two `CylinderGridVectors` with coordinates for the x - and y -direction. The vector class `CylinderGridVector` is a modified version of a class by Dominic Kempf [63], which contains several methods for conveniently adding successive coordinates according to different strategies, e.g. equidistant spacing, linearly increasing spacing, spacing increasing according to a geometric series, and many more. These methods are particularly useful in the present case of strongly varying mesh sizes in y -direction (cf. section 3.5.1.1), as they allow for a smooth transition of grid spacing between regions with very small and rather large mesh sizes. The method `generateTensorGrid()` calls these functions according to the

¹Intentionally, it is also a reference to the ACME (“A company that manufactures everything”) company from the TV series *Looney Tunes*.

minimum/maximum grid spacings specified in the config file in order to set up the tensor grid vectors X and Y . The generated vectors have the desired properties of a very fine resolution in y -direction at the membrane, and sufficiently coarse spacings away from the membrane and in x -direction, in order to minimize the number of grid points.

After generating the grid coordinate vectors, the actual grid hierarchy is set up. In particular, this hierarchy consists of a `Dune::YaspGrid` in a `Dune::GeometryGrid` metagrid, which itself is wrapped by a `Dune::MultidomainGrid`. The reason for this stacked metagrid hierarchy is the following: the structured `Dune::YaspGrid` is very efficient and it allows for an arbitrary parallel domain decomposition; however, it is an equidistant grid in each coordinate direction. To make this a tensor grid, the functionality of `Dune::GeometryGrid` is used, which enables to specify a coordinate function mapping the (equidistant) base grid nodes to those provided by the tensor product $X \times Y$. Finally, `Dune::MultidomainGrid` provides the possibility to define (even non-connected) subdomains and interior boundaries, which is just what we need for the multi-domain problem at hand. Each level in the grid hierarchy is paid for by an additional computational overhead. In performance tests, the runtime for a program that excessively called functions on the grid geometry – the most basic functions that are called in the innermost loops of the local operator, and therefore the most time-consuming operations introduced by the nested grid hierarchy – increased by approximately 30% when using `Dune::GeometryGrid`. We consider this acceptable with respect to the functionality provided. A similar overhead is expected for the `Dune::MultidomainGrid`.

Acme2CylFactory By implementing the (static polymorphism version of the) *factory pattern* [43], `Acme2CylFactory` creates an instance of an `Acme2CylPhysics` template class. Most of the template parameters of `Acme2CylPhysics` are specified by a static configuration class residing in the subdirectory `acme2_cyl/configurations`, containing compile-time constants and the class types for initial and boundary conditions. The resulting object of `Acme2CylPhysics` knows about the classes for initial and boundary conditions as well as all relevant model parameters for the desired simulation setup. The configurations classes, on the contrary, are only used as “read-and-forget” classes by `Acme2CylFactory`, i.e. their sole purpose is to provide read-only data for the instantiation of the physics class.

The reason for this approach is the restriction imposed by using generic programming, where all of the types used as template parameters must be known at compile-time. Therefore, for each of the configurations, there exists one

corresponding class. At compile-time, the code for each of these configurations is generated. This increases compilation time, but for a reasonably low number of configurations, this overhead is acceptable, as it adds the benefit that a certain simulation setup can then be chosen at runtime, by means of a config file parameter.

Acme2CylPhysics Once created, this class contains all relevant model parameters. It is the central data class in the application and provides additional methods for extracting data attached to certain grid cells and, most importantly, to both membrane interfaces. The values at the two opposite sides of the membrane are necessary for the calculation of the trans-membrane flux f_i^{memb} , as could be seen in section 3.4 on page 46. For this purpose, **Acme2CylPhysics** internally holds a map that identifies two opposite `Dune::IntersectionIterators` with each other, corresponding to the map $\mu(\mathbf{x})$ from section 3.4. For all the other data attached to grid entities, `std::vectors` are used in combination with `Dune::IndexSets`, which provide the corresponding index used in the vectors for a given grid element.

Since grid data and some meta information about the grid and its elements is needed virtually anywhere in the application, almost every one of the following classes hold a reference to the physics object. It is a bit unsatisfactory from the software design point of view to have such a central data class that is used everywhere in the program, but it is the only way if we do not want to sacrifice efficiency for a cleaner design.

Acme2CylGeometrySwitch This `struct` was added when extending the simulator from Cartesian 2D to cylinder coordinates. It seemed advantageous to be able to use both coordinate systems without having to maintain two different code bases or suffer from performance impairments, which could be achieved by using [template meta programs \(TMPs\)](#) [102], a technique that utilizes template specialization in order to implement conditional behavior depending on the type of an object. This is best exemplified when looking at the code:

```
struct Acme2CylGeometrySwitch
{
    template<typename GEO, bool useCylinderCoords =
        USE_CYLINDER_COORDINATES>
    struct GeometrySwitch
    {
        typedef GEO type;
    };
};
```

```
template<typename GEO>
struct GeometrySwitch<GEO,true>
{
    typedef Acme2CylGeometry<GEO> type;
};

//...
}
```

We see that the procedure is actually quite simple. `Acme2CylGeometrySwitch` contains a template `struct GeometrySwitch` with two template parameters: the type of the original (2D, Cartesian) `Dune::Geometry` class of the current grid, and a boolean flag that evaluates to `true` when the cylinder coordinate system is to be used. In the simplest case, the flag `useCylinderCoords` is `false` and thus the nested typedef `type` of `GeometrySwitch` will evaluate to the template parameter `GEO`. A second version of `GeometrySwitch` is partially specialized on the second template parameter. The compiler will give preference to this specialized version when the boolean flag is `true`, and the nested typedef `type` will evaluate to `Acme2CylGeometry<GEO>` instead of `GEO`. It is immediately clear that the flag has to be a compile-time constant in order for this to work. The default value `USE_CYLINDER_COORDINATES` can be set in a global header `constants.hh` for convenience. The functionality of `Acme2CylGeometry` is described next.

Acme2CylGeometry This class can be seen as a wrapper class around the original geometry class providing the transformation to cylinder coordinates. But technically, it works differently, since it is designed as a *mixin*:

```
template<typename Geometry>
class Acme2CylGeometry : public Geometry
{
    //...
};
```

The mixin pattern describes the method of deriving from a class which itself is given as a template parameter. It can be seen as the static polymorphism version of the famous Gang of Four *decorator* pattern [43]. Consequently, an `Acme2CylGeometry` object is not a wrapper around the original `Geometry`, it *is* a `Geometry` object and therefore inherits all of its methods. This way, overhead is only added for those member functions that need to be adapted. Furthermore, it

eliminates the overhead of additional lookups that would arise when using dynamic polymorphism, and it allows the compiler to perform better optimizations, as the father class type is known at compile-time. In conjunction with the switch `Acme2CylGeometrySwitch`, this can be implemented as follows:

```
// Get geometry object from a given entity iterator
// 'eit'
const GeometryOrig& geoOrig = eit->geometry();
// Use switch to choose original or cylinder geometry
// type, depending on compile-time flag
typedef typename Acme2CylGeometrySwitch::
    GeometrySwitch<GeometryOrig,useCylinderCoords>::
    type Geometry;
// 'geo' is now either a plain 2D or a cylinder
    geometry
const Geometry& geo(geoOrig);
```

When the flag `useCylinderCoords` is `true`, the 2D geometry `geoOrig` will be replaced by a new `Acme2CylGeometry` object `geo`, which is copy-constructed from `geoOrig`.

When it is `false`, the typedef `Acme2CylGeometrySwitch::GeometrySwitch<GeometryOrig>::type` will evaluate to the original type `GeometryOrig`, and the only overhead is the creation of a single `const` reference. With this setup we can switch between both coordinate systems simply by setting a single boolean compile-time flag. Finally, we can also create two different executables for each case from the same code, which allows for direct comparison between 2D and 3D results.

Acme2CylSetup The `setup()` method of this class essentially plugs together all the different DUNE components, particularly the local operators. Following the residual definition in section 3.5, one operator for the assembly of each residual in eqs. (3.7a), (3.7b), and (3.8) has to be defined.

In order to couple Nernst-Planck eq. (2.42) and Poisson eq. (2.43), however, the residuals for the spatial part of the subdomain Ω_{elec} are treated together by a single operator `Acme2CylOperatorFullyCoupled`, assembling the combined residual $R_{\Omega_{\text{elec}}} = (R_{\text{NP,S}}, R_{\text{P},\Omega_{\text{elec}}})^T$. Here, $R_{\text{P},\Omega_{\text{elec}}}$ denotes those entries of the full Poisson residual R_{P} from elements belonging to the Ω_{elec} subdomain. The class `Acme2CylOperatorFullyCoupled` is a modification of the `Dune::PDELab::ConvectionDiffusionFEM` local operator from `dune-pdelab`.

The modification involves the treatment of all $N + 1$ variables (N concentrations, 1 potential) in a single operator.

Using a single operator of convection-diffusion type is possible because both Nernst-Planck and Poisson equations can be written in the form of a convection-diffusion eq. (2.5). The two parameter classes `NernstPlanckParameters` and `PoissonParameters` are used to insert the correct coefficients for eq. (2.42) and eq. (2.43), respectively, into the convection-diffusion equation of form eq. (2.5).

The remaining entries $R_{P,\Omega_{\text{memb}}}$ of the full Poisson residual are assembled by a separate operator taking only contributions from the subdomain Ω_{memb} into account. For this, the existing class `Dune::PDELab::ConvectionDiffusionFEM` is used, again in conjunction with a `PoissonParameters` class.

A third operator `NernstPlanckTimeLocalOperator` handles the temporal part of the residual, $R_{\text{NP,T}}$. A combined instationary operator `Dune::PDELab::OneStepGridOperator` is obtained automatically by the functionality of `dune-multidomain`. This grid operator is handed over to the `Dune::PDELab::Newton` class, and all solver and infrastructure objects are forwarded to the class `Acme2CylSimulation`.

Acme2CylSimulation The simulation class is responsible for carrying out the time loop for the instationary problem. In each iteration, the new boundary conditions are calculated, especially the membrane flux, which is done in the helper class `MembraneFluxGridFunction`. Then the grid operator is used to assemble the full Newton matrix and residual, which is then solved for in each iteration according to section 2.2.3.

By a suitable choice of template parameters for the class `Dune::PDELab::MultiDomain::MultiDomainGridFunctionSpace`, the ordering of DOFs in the vector and matrix data structures can be specified. In conjunction with additional template parameters in `Dune::PDELab::ISTLVectorBackend`, DOFs can be *blocked* together. In this case, all unknowns belonging to a certain node are blocked together, yielding a matrix with $(N + 1) \times (N + 1)$ blocks that shows a block-tridiagonal pattern.

Note that this is only possible when using a single membrane element layer, since then the number of unknowns at each grid vertex is the same, i.e. N concentration and one potential variable. This is the precondition for using the vertex-blocking strategy, as the DUNE matrices currently only allow block matrices of equal size in the large system matrix. For $n > 1$ membrane elements, there are isolated potential variables at grid vertices inside the membrane, rendering the blocking

strategy impossible and prohibiting the solution of large systems, as the linear solver would not converge anymore.

The matrix pattern is illustrated in fig. 4.1: in comparison to a naive lexicographic order of DOFs as in fig. 4.1a, the vertex-blocked order in fig. 4.1b shows a much more advantageous diagonal structure, which reveals to consist of three block-diagonals in the zoom-in fig. 4.1c.

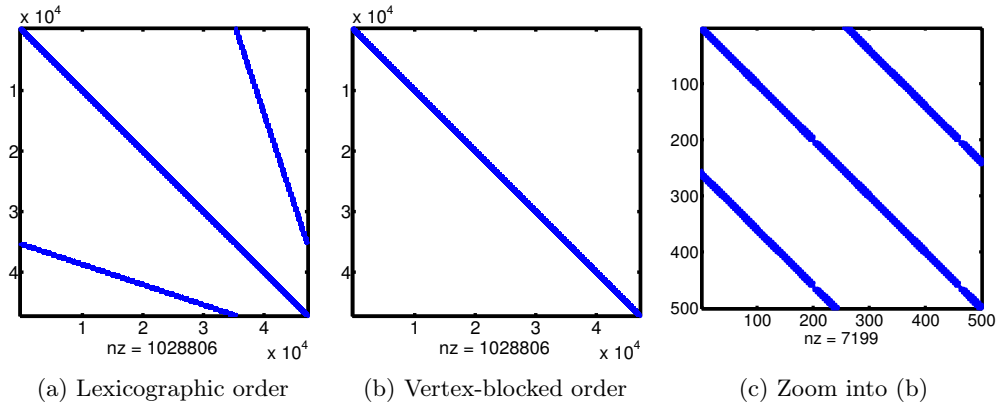


Figure 4.1: **Comparison of matrix structure depending on DOF ordering.** The left panel shows a lexicographic ordering of unknowns with an unfavorable sparsity pattern. Using a vertex-blocked ordering (*center*) yields an advantageous diagonal pattern, which reveals to consist of multiple block-diagonal matrices when zooming in (*right*).

This structure is very beneficial for the linear solver performance, as both `ISTL` implementations of the `ILU` and `AMG` preconditioners make great use of the block structure. For the iterative solver itself, i.e. `BiCGStab` or `GMRes`, the block-diagonal pattern is absolutely crucial, since otherwise it would not converge for setups with a large number of unknowns.

The termination criterion of the `Dune::PDELab::Newton` implementation depends on two values: the absolute norm of the residual, the *defect* $\|r^k\|$ in iteration k , denoted by `absLimit`, and the relative norm with respect to the initial residual $\frac{\|r^k\|}{\|r^0\|}$, the *reduction*. Tolerances for both errors can be provided, and convergence is acknowledged as soon as one of the error tolerances is satisfied.

Usually, prescribing a certain reduction is desired, as the absolute value of the defect depends on the mesh resolution. But in certain situations, providing an absolute limit is necessary, for example when the desired reduction can not be

reached in every time step due to changing dynamics (and hence the condition of the problem). One example for such a case is when a system reaches steady-state. Then, the initial defect will usually be small in each time step and using the reduction from previous time steps with higher initial defects would be too restrictive, as the initial defect now is already close to the reachable limit. These error tolerances have to be found individually for each problem, and they critically determine convergence and accuracy properties of the numerical solution.

After each solve of the `PNP` system, the simulation class triggers the output of solution vectors and diagnostic data in `Acme2Cy1Output`.

Acme2Cy1Output The output class essentially only writes out all relevant simulation data to files. However, the generic implementation is intrinsically quite complicated, as there is a plethora of PDELab grid functions living on different parts of the grid and on different function spaces. Therefore, not only do the grid function classes have different domain and field types, they also require different output strategies. For example, the membrane potential is a single value that is only defined on membrane interfaces, while the concentrations are aggregated into a vector of size N that lives on electrolyte elements only. This problem is solved again using `TMPs` [102]. The `TMP OutputStrategy` switches between the different output methods, based on the type of the grid function to be written.

Two file formats are used for the output: Gnuplot [104] and HDF5 [97]. For the gnuplot output, a custom output class `GnuplotTools2D` writes ASCII files that can be examined on-the-fly while the simulation is running. The adapter class `HDF5Tools` internally uses the HDF5 libraries to write the data to the complex binary `.h5` file format. In production runs, the gnuplot output is used only for the output of small diagnostic data, while the large solution vectors are written to the more storage-efficient HDF5 format.

5 Model of an Unmyelinated Axon

Things that are complex are not useful, things that are useful are simple.

(Michail Kalashnikov)

After the definition of the model in chapter 3 and its implementation in chapter 4, we are now ready to look at the results for an unmyelinated, homogeneous axon in extracellular fluid. Again, this chapter is based largely on an edited version of [82].

5.1 Simulation Setup

In the following, we consider a square computational domain of size $10\text{ mm} \times 10\text{ mm}$, where the axon extends from $y = 0$ to $y_{\text{memb}} = 500\text{ nm}$. Note that y_{memb} represents the axon radius due to cylinder symmetry. The membrane thickness was chosen to be $d_{\text{memb}} = 5\text{ nm}$.

The choice of grid parameters for the main setup used for most of the results in this chapter can be found in table 5.1. Cases with a different choice of parameters will be explicitly mentioned. In x -direction, a uniform spacing of h_x is used. The minimum Debye length for the given intra- and extracellular concentrations is about 0.9 nm , so a minimum grid spacing of $h_y^{\text{min}} = 0.5\text{ nm}$ was chosen at the membrane in y -direction to account for this. For the rest of the y -direction, a mixture of geometrically increasing and equidistant mesh widths was used: starting from the membrane, the grid spacing is smoothly increased up to a maximum of $h_y^{\text{max}} = 100\text{ }\mu\text{m}$. The large difference between these lengths underlines the multi-scale character of this model, resulting in a maximum anisotropy of $\frac{h_x}{h_y} = 200\,000$.

The diffusion coefficients D_i were chosen to be the diffusivity in water for each ion species. The relative permittivity ϵ was 80 in the electrolytes and 2 on the membrane, in accordance with [69]. The temperature was fixed at $T = 6.3\text{ }^\circ\text{C}$ as in the original HH model [55].

Table 5.1: Simulation parameters for the unmyelinated axon model.

Parameter	Value	Unit	Description
GRID			
x^{\max}	10	mm	Domain size (x -direction)
y^{\max}	10	mm	Domain size (y -direction)
y_{memb}	500	nm	Radius of the axon
d_{memb}	5	nm	Membrane thickness
h_x	100	μm	Mesh size in x -direction
h_y^{\min}	0.5	nm	Minimum mesh size in y -direction (Debye layer)
h_y^{\max}	100	μm	Maximum mesh size in y -direction
#DOFs	73 124	1	Total number of unknowns
PHYSICS			
ϵ_{memb}	2	1	Membrane permittivity
ϵ_{elec}	80	1	Electrolyte permittivity
$[\text{Na}]_0^{\text{CY}}$	12	mM	Intracellular Na^+ bulk concentration
$[\text{K}]_0^{\text{CY}}$	125	mM	Intracellular K^+ bulk concentration
$[\text{Cl}]_0^{\text{CY}}$	137	mM	Intracellular Cl^- bulk concentration
$[\text{Na}]_0^{\text{ES}}$	100	mM	Extracellular Na^+ bulk concentration
$[\text{K}]_0^{\text{ES}}$	4	mM	Extracellular K^+ bulk concentration
$[\text{Cl}]_0^{\text{ES}}$	104	mM	Extracellular Cl^- bulk concentration
D_{Na}	1.33×10^{-9}	m^2/s	Na^+ diffusivity
D_{K}	1.96×10^{-9}	m^2/s	K^+ diffusivity
D_{Cl}	2.03×10^{-9}	m^2/s	Cl^- diffusivity
g_{Na_v}	120	mS/cm^2	Conductance of the voltage-gated Na channel
g_{K_v}	36	mS/cm^2	Conductance of the voltage-gated K channel
g_L	0.5	mS/cm^2	Total leak conductance
NUMERICS			
reduction	1×10^{-5}	1	Newton reduction
absLimit	1×10^{-5}	1	Newton absolute limit
t_{end}	20	ms	Simulated time
Δt_{min}	0.05	μs	Minimum time step
Δt_{max}	50	μs	Maximum time step

Parameter	Value	Unit	Description
NUMERICS			
$\Delta t_{\max, AP}$	10	μs	Maximum time step during AP
it_{\min}	10	1	Newton iteration threshold for time step increase
it_{\max}	30	1	Newton iteration threshold for time step decrease

5.1.1 Parallelization

For the simulation parameters summarized in table 5.1, the problem consists of 73 124 unknowns per time step. When simulating until $t_{\text{end}} = 20$ ms, we obtain an average time step size of $13.6 \mu\text{s}$, resulting in a total computation time of about 19 h for the sequential solution on a single processor.

While this is not intractably long, it is certainly not suitable for rapid prototyping. Therefore, it seemed beneficial to parallelize the algorithm, also considering that we planned to add myelin to the model, which would supposedly require a much higher number of DOFs.

In an effort to minimize problems due to an improper domain decomposition, the grid was chosen to be partitioned only in x -direction. This way, the membrane will only be cut vertically, as suggested in fig. 5.1, where the processor boundaries are marked by vertical dashed lines. This partition ensures that the two electrolyte subdomains do not get separated from their associated patch of membrane, which would cause problems implementation-wise for the calculation of membrane fluxes.

It also prevents numerical problems, as membrane and Debye layers should be handled on a single processor to cope with the grid anisotropy. An overlap of one cell is used at processor boundaries, marked by the shaded area in fig. 5.1.

To assess the performance of the parallelization, we ran simulations on the same problem with different processor counts (*strong scaling*). Table 5.2 shows the timings for different processor counts. For $p = 10$ processors the total computation time is reduced to about 2.5 h, yielding a speedup of 7.58.

These results are quite good, considering that, according to the developers¹, the stacked metagrid hierarchy described in section 4.2.2 can not be expected to scale perfectly in parallel. In addition, we have to consider that even in the parallel

¹i.e. DUNE developers Steffen Müthing for `Dune::MultidomainGrid` and Christian Engwer for `Dune::GeometryGrid` (personal communication)

runs, quite a large amount of debug output is written to disk and standard output sequentially by the root node, which further impairs the scaling performance.

Since the goal of the parallelization was not getting perfect scaling properties, but rather achieving a reasonable speedup in computation time, we were content with these results and no further attempt was made to specifically optimize the algorithm to yield a better parallel performance.

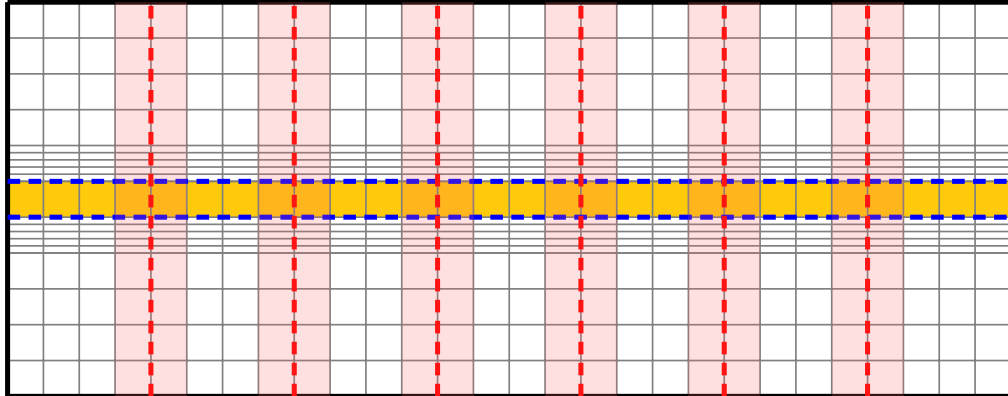


Figure 5.1: **Partition of the unmyelinated axon computational domain for the parallel case.** The computational domain with its three subdomains is shown as before; additionally, processor boundaries (*vertical dashed lines*) and overlap elements (shaded) are shown, in this case exemplary for $p = 7$ processors. Note that this method gives optimal control over the load balancing and ensures that membrane interfaces never coincide with processor boundaries.

5.1.2 Linear Solver and Numerical Performance

The number of 73 124 DOFs is a small one within the context of HPC, and this would allow for the usage of a direct [linear solver \(LS\)](#) like SuperLU on each processor, even in the sequential case. Nevertheless, we chose to use an iterative solver in perspective of solving larger problems. For the present problem, an overlapping [BiCGStab](#) solver preconditioned by an [ILU0](#) decomposition proved to be a good choice, as visible in the solver statistics table [5.3](#).

Only one or two linear solver iteration per Newton iteration were needed on average, indicating that the [ILU0](#) preconditioner is very effective. This is not surprising, since we deliberately chose to group together all unknowns at a certain

Table 5.2: **Simulation timings for the parallel solution using different processor counts p .** For each processor count p , the total computation time, the needed number of time steps, and the average solution time per time step (full Newton iteration) are shown together with the resulting speedup with respect to the sequential problem.

p	Total comp. time [s]	# time steps	avg. time / time step [s]	speedup
1	69042.3	1469	47.00	
2	35556.1	1469	24.20	1.94
4	18721.3	1469	12.74	3.69
10	9101.12	1469	6.20	7.58

grid vertex in order to create dense 4×4 blocks. This arrangement allows the [ILU0](#) preconditioner to invert the blocks on the diagonal *exactly*, which essentially captures the nonlinearity in the [PNP](#) system, the coupling between Nernst-Planck and Poisson equations.

Table 5.3: **Solver statistics per time step for the parallel solution using different processor counts p .** All values are averages over all time steps. The total time includes matrix and residual assembly as well as the actual solution time by Newton’s method. The solver time includes both [ILU](#) decomposition and [LS](#) time, the [LS](#) time only the actual [BiCGStab](#) solve. The number of linear solver iterations per Newton iteration stays at a very low level for all processor counts.

p	speed-up	total time	assembler time [s]	solver time [s]	LS time [s]	LS it.	LS time / it. [s]	Newton it.
1		44.9519	38.0977	0.30445	0.085314	1.22	0.070337	1.9993
2	1.94	23.1669	19.4495	0.34784	0.137967	2.3	0.059875	1.9993
4	3.69	12.1984	10.2244	0.22298	0.095068	2.44	0.043907	1.9993
10	7.55	5.9506	4.8945	0.27284	0.128984	2.8	0.046578	1.9993

5.2 Simulation Results

The remaining part of this chapter serves to show several simulation results for the case of an unmyelinated axon. The most interesting case of an axon firing an action potential and demonstrating the various processes involved in this state of excitation, however, requires a valid initial state, which is the resting state of the membrane.

Since the concentration distribution at resting potential as well as the precise potential profile are unknown, a transient *equilibration* simulation has to be carried out for each simulation setup with a change in the computational grid, boundary conditions, or bulk concentration values n_i^0 . The process of obtaining a physically consistent equilibrium state is described first, followed by the actual action potential simulations.

5.2.1 Equilibrium States

As quoted in section 2.3.3.1 on page 31, the resting state of the neuronal membrane is not a true equilibrium state in the thermodynamical sense, as only the *sum* of fluxes goes to zero. Nevertheless, we will use the terms *resting state* and *equilibrium state* synonymously, which is also common practice in the literature. The reader should keep in mind that indeed the more general concept of a *flux equilibrium* is meant by this.

The procedure of obtaining the equilibrium state is as follows: the model from section 3.2 is initialized by setting the ion concentrations within one electrolyte domain uniformly to their intra- and extracellular bulk values n_i^0 (see table 5.1). Then the leak channels are opened and the evolution of membrane currents is simulated using a fixed time step $\Delta t = 10 \mu\text{s}$ until the sum of trans-membrane fluxes is sufficiently close to zero, which, for most setups, was the case after a simulated time of 1 s. It is important to note that voltage-dependent channels are kept closed during the whole equilibration to avoid premature AP firing.

The bulk concentrations were chosen such that each electrolyte initially is electroneutral, i.e. the net charge is zero, which is a reasonable assumption both physically and biologically with respect to energy minimization principles.

In the following, the generated equilibrium states are depicted by plotting the relevant values along a line perpendicular to the membrane, i.e. parallel to the y -axis. This is possible because the x -components of the ion fluxes during equilibration are negligible, therefore the solution only changes notably in y -direction. This fact allows for a fast equilibration procedure: the equilibrium state is obtained for a grid with only one element in x -direction, and the hereby

obtained values are interpolated onto the fine grid when starting the actual AP simulations. This way, the equilibrium state can be generated within a few minutes instead of hours.

When selectively opening only the leak channel for one ion species, the equilibrium membrane potential is expected to be equal to the corresponding ionic reversal potential, as predicted by eq. (2.27). The first two rows in table 5.4 show the calculated equilibrium potentials which indeed match the value calculated by Nernst's equation.

When opening both Na and K leak channels, the equilibrium membrane potential will reach a value between the two channels reversal potentials, depending on the ratio of Na and K leak conductances. The relative leak conductances that result in a resting potential of about -65 mV can be found in the third row of table 5.4. The resting potential exactly matches the value predicted by the PCM eq. (2.29). The total leak conductance (0.5 mS/cm²) was always kept constant.

Table 5.4: **Relative leak conductances and resulting equilibrium membrane potentials.**

Leak conductances		Equilibrium membrane potential
g_{Na_L}	g_{K_L}	$[[\phi]]$
1.0×0.5 mS/cm ²	0.0×0.5 mS/cm ²	50.62 mV
0.0×0.5 mS/cm ²	1.0×0.5 mS/cm ²	-82.18 mV
0.13×0.5 mS/cm ²	0.87×0.5 mS/cm ²	-64.92 mV

Figure 5.2 shows the intra- and extracellular charge density profile at equilibrium. As predicted by the Poisson-Boltzmann eq. (2.22), both electrolytes adjust their concentrations to follow a Boltzmann distribution towards the membrane. Figure 5.3 shows the evolution of membrane fluxes during the equilibration phase. The sum of inward- and outward-directed fluxes tends to zero, marking the neuron's resting state.

5.2.2 Action Potential

The action potential simulations are carried out by loading the equilibrium state generated as described in the previous section. Then the gating particles p of

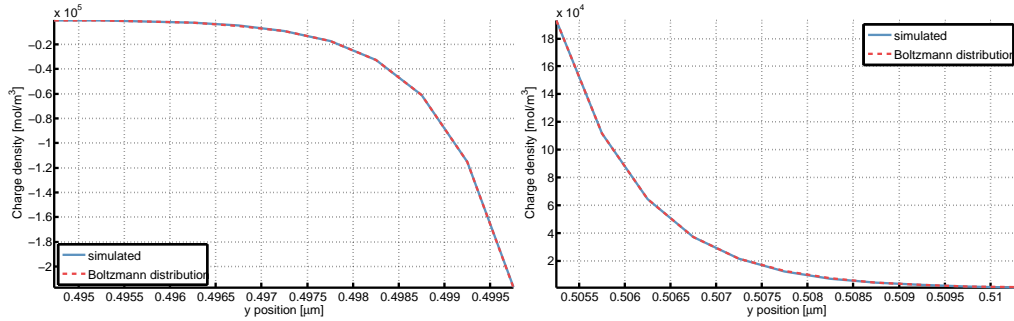


Figure 5.2: **Equilibrium charge density.** The equilibrium charge density resulting from contributions of all three ion species (*solid lines*), compared with a Boltzmann distribution (*dashed lines*). Only the range close to the membrane is depicted, where the charge density profile undergoes its greatest change.

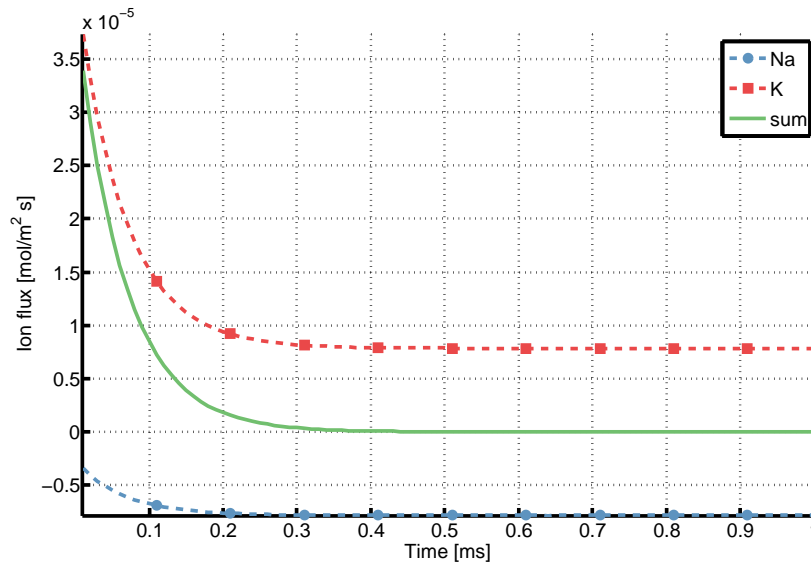


Figure 5.3: **Equilibration of membrane fluxes.** Na (*lower line*), K (*upper line*), and summed membrane fluxes (*middle line*) are shown for the equilibration phase. The sum of fluxes vanishes at equilibrium.

voltage-gated channels are initialized to their steady state

$$p_{\infty}(V_{\text{rest}}) = \frac{\alpha_p(V_{\text{rest}})}{\alpha_p(V_{\text{rest}}) + \beta_p(V_{\text{rest}})}$$

with respect to the calculated membrane resting potential V_{rest} , as in [65, p. 146]. This is also the default initialization procedure in NEURON [53, chapter 8.3]. As opening the voltage-gated conductances changes the ratio of conductances between ion species, the leak channel ratio is corrected such that the ratio of each ion species' sum of conductances with respect to the total membrane conductance stays constant, corresponding to the values chosen for the equilibration procedure (cf. table 5.4). This ensures that the membrane potential $[\phi]$ does not drift off from the generated resting potential V_{rest} after opening the voltage-gated channels.

Here we acknowledge again that only *relative* conductances matter for the membrane potential, as can be seen both from the Goldman eq. (2.28) and the PCM eq. (2.29), i.e. the general validity of the equilibrium state is maintained by this modification. However, the recalculation of leak conductances involves rounding errors, which have a small, but observable effect on the membrane potential in the sub-millivolt range. The system is allowed to settle with respect to the changed channel conductances for a low number of time steps to account for this.

To evoke an action potential, a sodium rectangle pulse is injected into the cell by adding a fixed amount of sodium at the stimulation site located near the left domain boundary at $\mathbf{x}_{\text{stim}} = (150 \mu\text{m}, 0 \mu\text{m})$ for 2 ms. The pulse had a value of 0.965 nA in this setup.

The membrane is depolarized close to the stimulation site and, after reaching threshold, an action potential is generated due to the ion channel kinetics from section 2.3.3.2. The potential wave travels along the axon, opening more channels along the way and keeping the action potential alive, resulting in a wave traveling at constant velocity. The conductance velocity depends on the time constants of the ion channel kinetics, but also on the intra- and extracellular ion diffusion coefficients, and has a value of about 0.93 m/s for this setup.

5.2.2.1 Intracellular Potential

Figure 5.4 shows the potential time courses at different x -positions along the axon. The y -position is about 488 nm, but this does not have a large impact, as the intracellular potential is fairly constant outside the Debye layer in this direction.

In a first approximation, it is also equal to the membrane potential, since the extracellular potential is much smaller. The first AP has a higher amplitude than the following ones, caused by the proximity to the stimulation site. Also, switching off the stimulus is reflected by an artifact in the repolarization phase of the first AP at $t = 2$ ms.

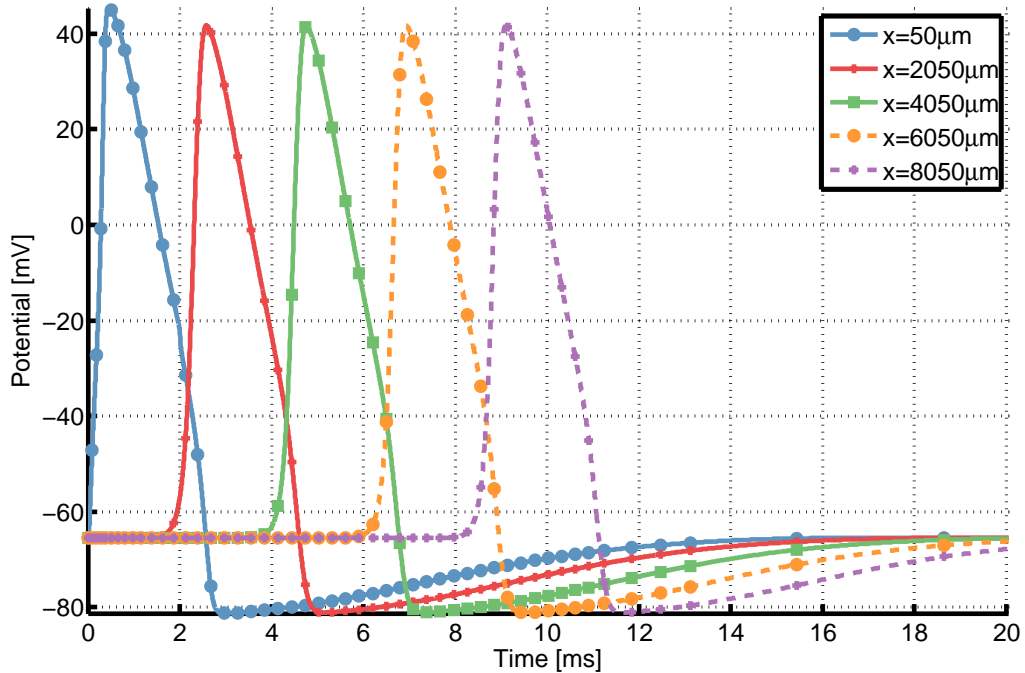


Figure 5.4: **Action potentials evoked at equidistant positions along the axon.** The leftmost AP has a higher amplitude than the others due to the vicinity to the stimulus site. The following curves at equidistant positions along the axon are identical and show constant onset delay, indicating a wave traveling with constant speed.

5.2.2.2 Membrane Flux

The total membrane flux consists of two main components: an ionic and a capacitive component, see fig. 5.5. The ionic flux itself is the sum of sodium and potassium fluxes through the membrane's (active and passive) ion channels. The other component is a consequence of the electric properties of the membrane. As seen in section 2.3.3, the membrane separates charges and therefore acts as a

capacitor. The capacitive flux is a consequence of the virtual current caused by charge redistribution at both sides of the membrane. In cable equation models, this current has to be specified explicitly by setting a membrane capacitance parameter, while in the electrodiffusion model, it is implicitly contained in the PNP equations through the concentrations contained in the adjacent electrolytes. It can be calculated by using the textbook formula for the capacitive current

$$I_C = C \frac{dU}{dt} = C \frac{kT}{e} \frac{d[\phi]}{dt}, \quad (5.1)$$

where U is the potential across the capacitor, in this case the membrane potential $[\phi]$ brought to units of Volts, and C is the membrane capacitance.

In the following, C will be expressed as the capacitance per unit area, as it is also the common choice in electrophysiology literature. It can be calculated by the formula for a cylinder capacitor

$$C = 2\pi\epsilon_0\epsilon_{\text{memb}} \frac{l}{A \ln \frac{r_2}{r_1}} \approx 0.35 \times 10^{-2} \frac{\text{F}}{\text{m}^2} \quad (5.2)$$

with l the length, A the surface area of the membrane patch, and $r_1 = y_{\text{memb}}$ and $r_2 = y_{\text{memb}} + d_{\text{memb}}$ marking the opposite membrane boundaries, respectively. The value turned out to be the same when using the formula for a parallel-plate capacitor

$$C = \frac{\epsilon_0\epsilon_{\text{memb}}}{d_{\text{memb}}}, \quad (5.3)$$

suggesting that the membrane can be regarded as a parallel capacitor in a first approximation, as the membrane thickness is small compared to the axon diameter. In summary, we have

$$f_C = \frac{1}{eN_A A} I_C = \frac{1}{eN_A A} \frac{\epsilon_0\epsilon_{\text{memb}}}{d_{\text{memb}}} \frac{kT}{e} \frac{d[\phi]}{dt}$$

as the third trans-membrane flux next to the ionic fluxes f_{Na} and f_{K} defined in section 3.4 on page 46.

The total ionic flux follows the sodium flux in the rising phase of an AP and is later antagonized by the potassium flux, resulting in a “down-up” shape in fig. 5.6a. One interesting detail here is the small peak at the rear end of the ionic flux. This comes out of the standard HH model directly, because the sodium current declines faster than the potassium current. This feature has been referred

to as a “gratuitous bump” in [31, p. 307] and can be interpreted as an artifact of the original HH model, although later studies showed that such a structure can be observed under the influence of certain drugs in experiments.

In contrast, the capacitive flux is proportional to the time-derivative of the membrane potential and therefore shows an opposed behavior in fig. 5.6b. The sum of both components results in the total membrane flux in fig. 5.7, following roughly a triphasic “up-down-up” shape.

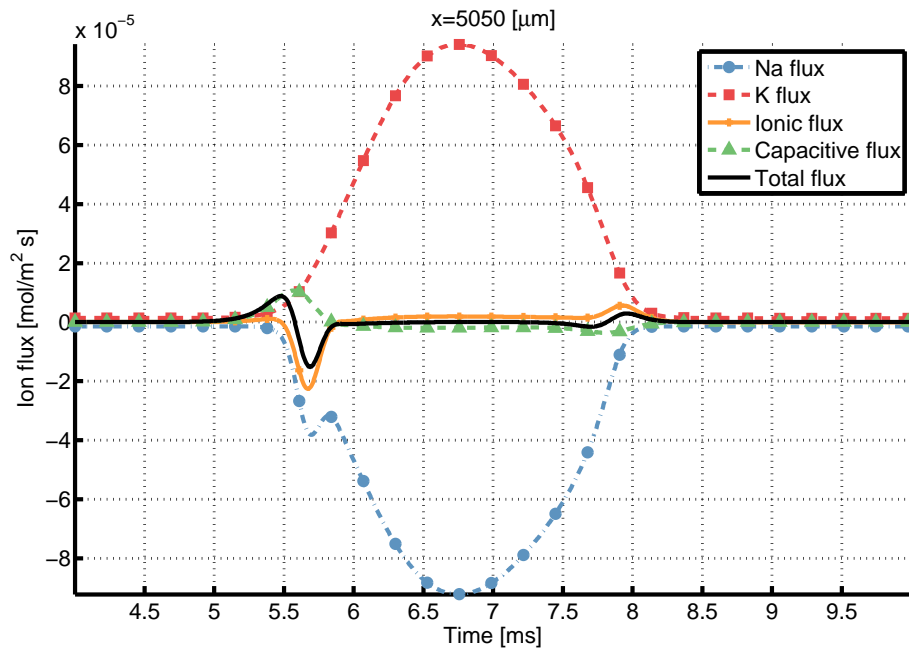


Figure 5.5: **Different components of the membrane flux at fixed point on the membrane.** These are shown at the same x -coordinate as the potential curves from fig. 5.8. The total flux consists of capacitive and ionic flux, which itself is the sum of Na and K channel fluxes.

5.2.2.3 Near- and Farfield Extracellular Potential

We will now focus on the time evolution of EAP signals at any point in the extracellular domain. In fig. 5.8, the potential time courses are plotted for the same x -coordinate at increasing distances from the membrane. Some major features can be identified from these curves: a first positive peak (P1) followed by a larger negative peak (N1), then a (very) small second positive peak (P2) with a

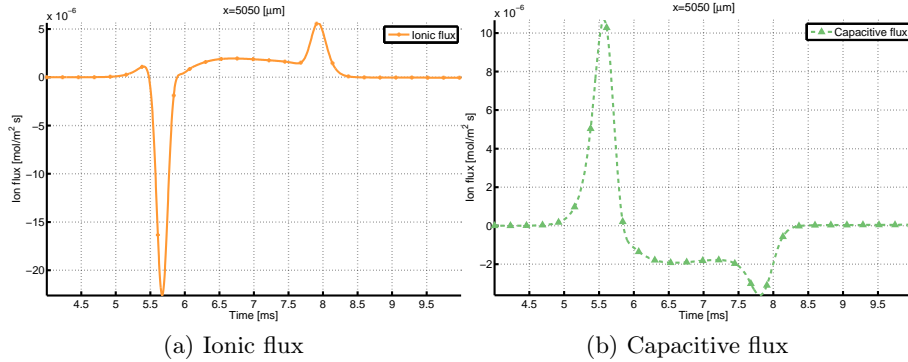


Figure 5.6: **Ionic and capacitive membrane flux at fixed point on the membrane.** Shown are the respective components of the total membrane flux from fig. 5.5. The ionic flux is dominated by the sodium current during the depolarization phase, and later by the potassium current during repolarization; therefore it has a “down-up” shape. On the contrary, the capacitive flux is proportional to the time derivative of the membrane potential, therefore it goes up before it goes down. Together, these two components form the “up-down-up” shape of the total membrane flux in fig. 5.5.

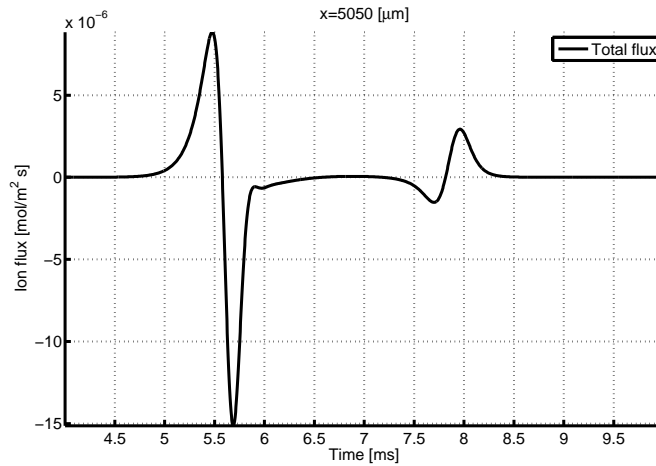


Figure 5.7: **Total membrane flux at fixed point on the membrane.** This is the same total flux as in fig. 5.5, shown separately.

subsequent longer phase of slowly varying potential with negative curvature (S), and a last peak (P3). This characteristic “up-down-up” shape is maintained at various distances from the membrane. The potential time course generally shows similarities with the total membrane flux at the same x -coordinate (fig. 5.5), suggesting that the membrane currents are the main contributors to the EAP.

The EAP of the point closest to the membrane, however, shows deviations from the general pattern, notably in the rear part P2 – S – P3. Consequently, the second peak P2 does not look like a peak, but more like a kink at a distance of only a few micrometers away from the membrane, because the following part S shifts from a negative edge towards a more or less constant, slightly negatively curved bow. We will see later on that this can be attributed to electrostatic forces from the membrane and the resulting concentration redistributions influencing the nearfield potential.

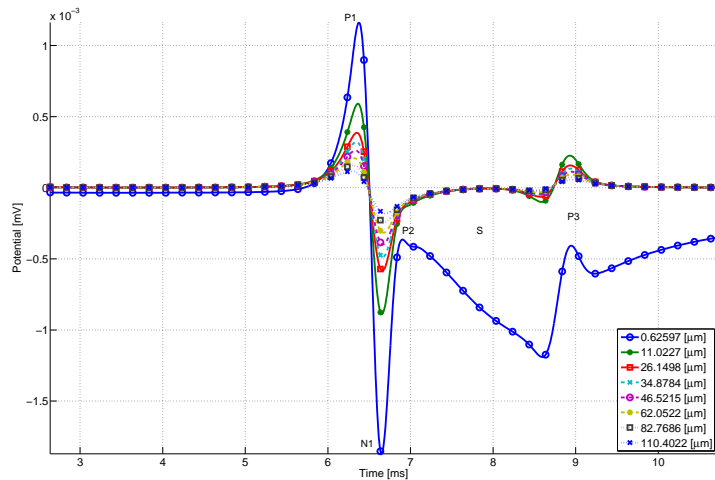


Figure 5.8: **Extracellular potential at various distances from the membrane.** The potential time courses for a fixed x -coordinate $x = 5.05$ mm are shown for several different y positions. While the characteristic triphasic shape is maintained, the amplitude spans several orders of magnitude as the signal is strongly attenuated with distance.

Since the action potential is a traveling wave, we can alternatively look at snapshots of the extracellular potential and concentrations at a fixed point in time. These profiles simply move through space with a constant (known) velocity, such that the complete information about the potential and the EAP dynamics at any point in space can be gained from these plots. As the signal moves in

positive x -direction for all following snapshots, we read from “right to left” when assessing the behavior of the profile in time, as opposed to the time course in fig. 5.8.

As expected, the nearfield potential profile (a few nanometers to about $10\ \mu\text{m}$ from the membrane) in fig. 5.9 shows the same pattern as the total membrane flux in fig. 5.7 and the EAP in fig. 5.8. The profile begins to the right with a rise in the potential (corresponding to P1) followed by a sharp drop (N1) and another rise (P2). After this first phase, the potential has a longer phase of low variation (S) until another, less pronounced peak is observed at the rear end of the traveling action potential (P3).

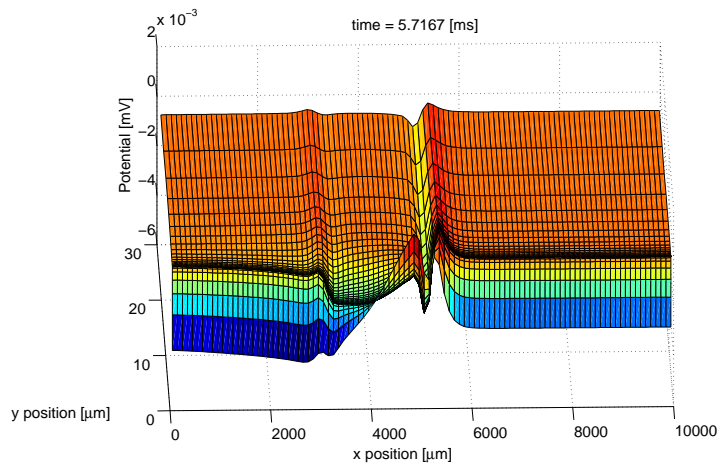


Figure 5.9: **Snapshot of the nearfield extracellular potential profile.** The plot shows the potential values for a stripe of the extracellular domain Ω_{ES} just above the extracellular Debye layer (i.e., above the gray area from fig. 3.1), at a fixed time. It shows a more complex structure compared to the Debye layer.

The distance between the end of P1 and the beginning of P3 at the beginning of the nearfield region is a good measure for the timescale of the extracellular field we term the “EAP valley length”, the region with negative potential values. It gives a characteristic length scale for the range of simultaneous ion channel activity along the axon. In this simulation, it is about $2000\ \mu\text{m}$ at the beginning of the nearfield region. Of course, the EAP valley length is largely determined by the AP velocity.

Figure 5.10 shows the potential for a large part of the extracellular space on a logarithmic scale, demonstrating that the nearfield potential profile essentially

continues into the farfield, albeit attenuating quickly with distance. The details of the shape that could be previously seen in figs. 5.8 and 5.9 have been smoothed out in this depiction by the logarithmic scaling, but the general pattern of the EAP stays the same: a positive upwind domain (P1) just in front of the opening channels, followed by a negative middle region (N1, S), and then again a positive rear domain (P3). The EAP valley length (the diameter of the green area) increases notably with distance, which we account to the diffusive character of the system.

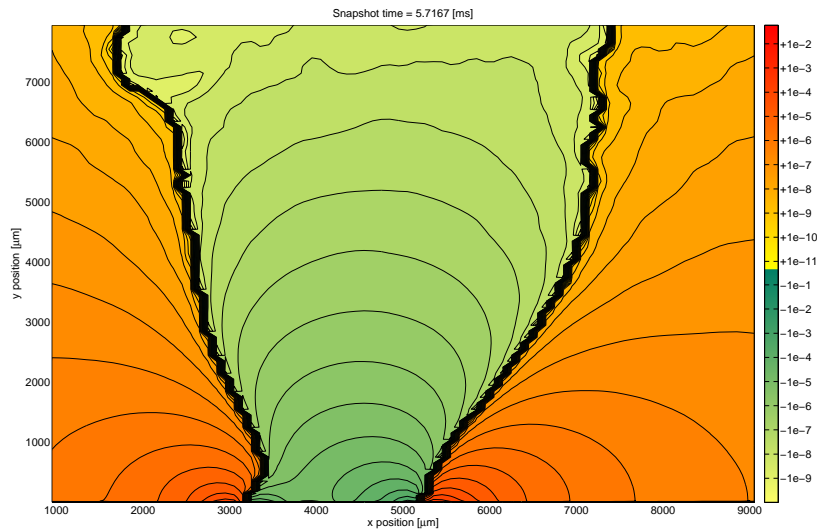


Figure 5.10: **Snapshot of the whole extracellular potential.** A contour plot of the log-scaled extracellular potential at a fixed time point. The domain was cut at the left, right, and upper boundary to exclude artifacts introduced by the boundary where the potential is almost zero and switches signs due to small numerical errors.

5.2.2.4 Debye Layer Extracellular Potential

The spatial structure of the extracellular potential profile within the Debye layer, i.e. at most a few nanometers from the membrane, shows a different picture, see fig. 5.11a. It is almost exclusively dominated by the intracellular potential (cf. fig. 5.11b), meaning that the intracellular potential spreads across the membrane into the extracellular space, if with a greatly reduced amplitude. We call this potential the “echo” of the action potential.

The potential damping over the membrane will now be quantified. In [67, chapter 12], a parameter s is used to estimate the electrical coupling between intra- and extracellular electrolytes. It is defined as

$$s = \frac{\gamma_D}{d_{\text{memb}}} \frac{\epsilon_{\text{memb}}}{\epsilon_{\text{elec}}} \quad (5.4)$$

in our usual notation of lengths and permittivities, and with γ_D being the Debye length of the extracellular electrolyte. For $s = 1$, the electrolytes are fully coupled, for $s = 0$ fully decoupled, meaning that the extracellular side does not see anything of the intracellular potential ($\phi_{\text{out}} = 0$). In the present setup, $s \approx 0.0045$. This is small, but not at all negligible, since extracellular potentials are generally orders of magnitude smaller than intracellular ones. As a result, a small s might still yield a potential that is comparable in amplitude to the extracellular potential, in absolute numbers. A value of $s = 0.0045$ describes two weakly coupled electrolytes.

Moreover, we found that s is not only a vague estimation for the coupling, it is indeed the *exact* constant of proportionality between intra- and extracellular potential at the membrane interface:

$$\phi_{\text{out}} = s\phi_{\text{in}} . \quad (5.5)$$

We found this relation to be very accurate over the whole time course of a simulation for an arbitrary x -coordinate along the membrane, implying that the Debye layer profile in fig. 5.11a is just the constant s times its intracellular counterpart in fig. 5.11b. As a result, the membrane potential can be expressed in terms of only one of these two potentials:

$$\begin{aligned} \llbracket \phi \rrbracket &= \phi_{\text{in}} - \phi_{\text{out}} = \phi_{\text{in}} - s\phi_{\text{in}} = (1 - s)\phi_{\text{in}} \\ &= \frac{1 - s}{s}\phi_{\text{out}} . \end{aligned}$$

This means we have found an analytical expression for the relation of the electric potentials at opposite membrane boundaries, provided only a few physical parameters of membrane and electrolyte are known. It remains to be shown under which conditions this relation is valid. In extreme cases – where the bulk electrolyte ceases to exist, as the bulk concentrations are not constant anymore (see also chapter 7) – we found that eq. (5.5) does not hold anymore. Under most physiological conditions, however, it might be a valid relationship that would

enable the calculation of the extracellular membrane interface potential from the intracellular one and vice versa.

Looking at the definition of s , another interesting connection can be made. An extracellular solution with higher ionic strength in the sense of eq. (2.26) implies, according to eq. (2.25), a smaller Debye length, which, in turn, means faster screening of concentrations. As a result, also the constant s is decreased, lowering the electrical coupling with the intracellular electrolyte. Essentially, this corresponds to a local increase in permittivity. It can be understood by recognizing that the concentrations always counteract the membrane potential – as counterions are attracted and co-ions repelled – such that a stronger electrolyte with a higher concentration of counterions will be more successful in “swallowing” the membrane charge and therefore screening the potential.

The question of how much of the intracellular potential is “felt” on the extracellular side is hence not only a property of the membrane, but to a large degree also of the extracellular electrolyte itself. This fact raises the question whether the assumption of fully independent intra- and extracellular electrolytes – which some models rely on, see below – can be made in general. We now see that the degree of dependence heavily relies on the magnitude of concentrations, which might vary significantly among different brain areas.

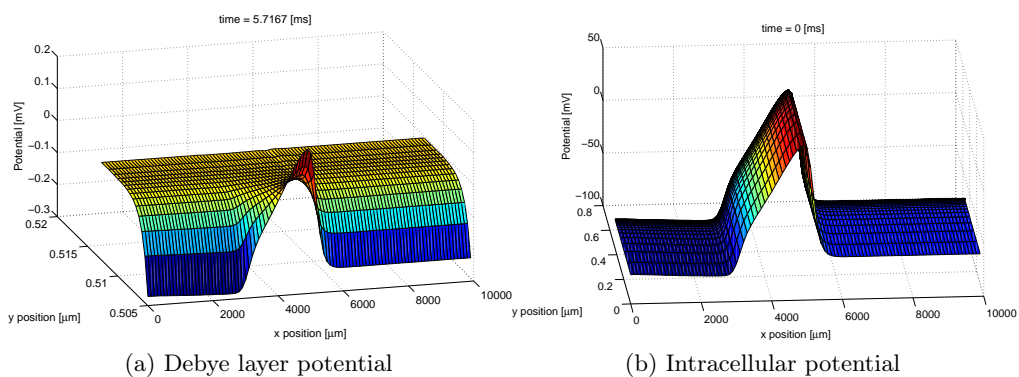


Figure 5.11: **Snapshot of the Debye layer extracellular potential profile.**

In the left part, the potential values for a narrow stripe of the extracellular domain Ω_{ES} (corresponding to the gray area from fig. 3.1) just above the membrane are plotted at a fixed time. This turns out to be the intracellular action potential (right) propagating over the membrane, with a significantly reduced amplitude.

5.2.2.5 Transition Interval from Debye Layer to Nearfield

We have seen that nearfield/farfield and Debye layer potential time courses behave quite differently. While the former is dominated by membrane currents, the latter is driven by the intracellular potential, damped by a constant factor s . To further investigate the change of **EAP** shape during the transition between Debye layer and nearfield, we look at extracellular time courses much in the same way as in fig. 5.8, but this time only for positions located between Debye layer and nearfield, in fig. 5.12. The transition between the **AP** echo shape very close to the membrane (figs. 5.12a and 5.12b) and the more intricate shape following the membrane flux (figs. 5.12f to 5.12i) can be observed to happen within the range of only a few nanometers, as depicted in figs. 5.12c to 5.12e.

In the course of this transition, the single peak from fig. 5.11a we termed the **AP** echo is divided into two parts P1 and P2 by an interrupting negative peak N1. Comparing again with fig. 5.5, we see that N1 results from the negative peak present in the membrane flux, which is the consequence of opening voltage-gated sodium channels and the following massive depletion of sodium ions. Roughly speaking, the activation of sodium channels and the resulting negative peak N1 splits up the single positive peak from the **AP** echo into two peaks P1 and P2. Looking closely, we can also see that the interrupting negativity N1 time-shifts from left to right between fig. 5.12c and fig. 5.12f, shifting also the relation in magnitude between P1 and P2. Furthermore, the rear part of the **AP** is superimposed by the membrane flux components S and P3.

This interesting behavior can be further elucidated by looking at the large concentration and potential gradients present in this range, illustrated for the potential in fig. 5.13. It shows the spatial profile of the **EAP** time courses from fig. 5.12 – normalized to baseline at $t = 2$ ms in each point – at distinct time points, which are indicated as vertical lines in fig. 5.12. The markers in fig. 5.13, on the other hand, correspond to the respective positions of figs. 5.12a to 5.12i.

We can see a rapid fall-off in the potential profile with increasing distance from the membrane in the normal scale fig. 5.13a, but the log-scaled fig. 5.13b shows a switch in fall-off behavior. Note that there are some irregularities in the transition interval where the **EAP** shape changes. From this, we can read that the potential time courses are not directly comparable at the chosen time points. For example, peak P1 in fig. 5.12c is smaller than P2, whereas in fig. 5.12d the opposite is the case, due to the time-shift of N1 over distance. A better approach to define the potential decay is via its maximum value in time. This is illustrated in fig. 5.14, which shows the peak values of the potential time courses from fig. 5.12 (again

5 Model of an Unmyelinated Axon

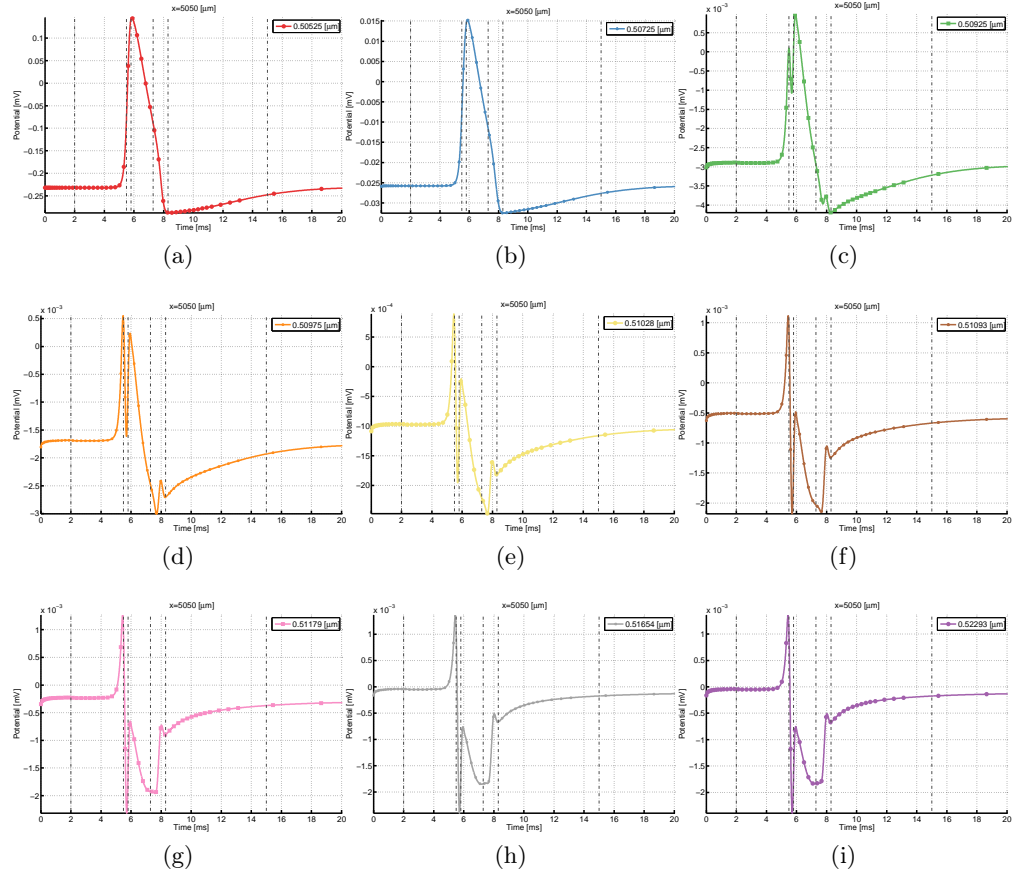


Figure 5.12: **Extracellular potential time courses over Debye layer.** For a selected number of points in the Debye layer, the extracellular time courses show a transition from the AP echo dominated shape directly at the membrane (panels a-b), via an intermediate shape (panels c-e) towards the characteristic up-down-up shape (panels f-i), which is also found at more distant positions, see fig. 5.8

relative to baseline). The peak for each point possibly occurs at a different point in time.

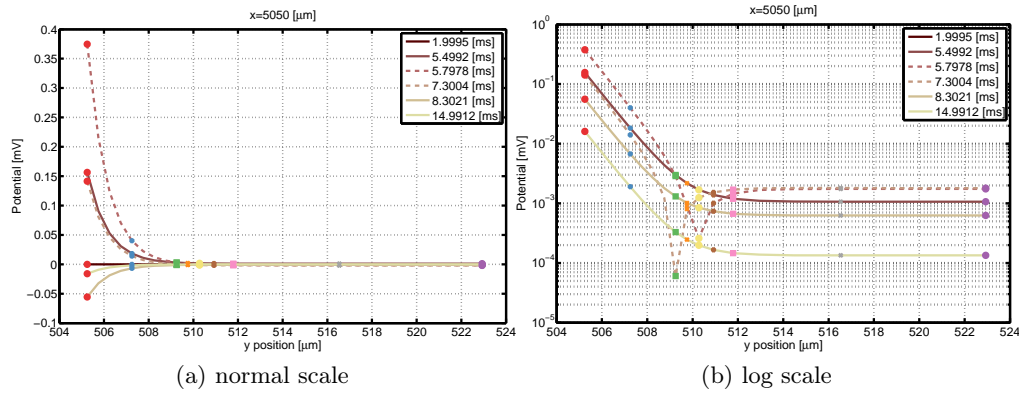


Figure 5.13: **Potential profile over Debye layer at various time points.**

The values are relative to baseline at $t = 2$ ms in each point. The left panel shows the potential profile close to the membrane, which is attenuated quickly with distance, independent of its value at the extracellular membrane interface at different times. The right panel shows the log-scaled absolute values of the same data, demonstrating the exponential screening over the Debye layer. Since the potential courses are not directly comparable due to the change in shape over the Debye layer, the log-scaled plot shows some irregularities in this region.

The analog is shown for the charge density in fig. 5.15. As before, we see a rapid fall-off in normal scale plot (fig. 5.15a), which can be shown to be exponential with an appropriate logarithmic scaling (fig. 5.15b).

In contrast to the charge density, a significant change in fall-off behavior is found in the potential when traversing the Debye layer. Over the first 5 nm from the membrane, it drops exponentially together with the charge density. Then the rapid decay changes to a much slower one, where it is driven primarily by membrane currents whose influence does not drop that drastically with distance. Thus, instead of dropping towards zero as the concentrations do, the potential only drops exponentially until it has reached a value in the same order as the membrane current contributions.

We conclude that close to the membrane, the influence of the intracellular potential is so large that all contributions by membrane fluxes are hidden in

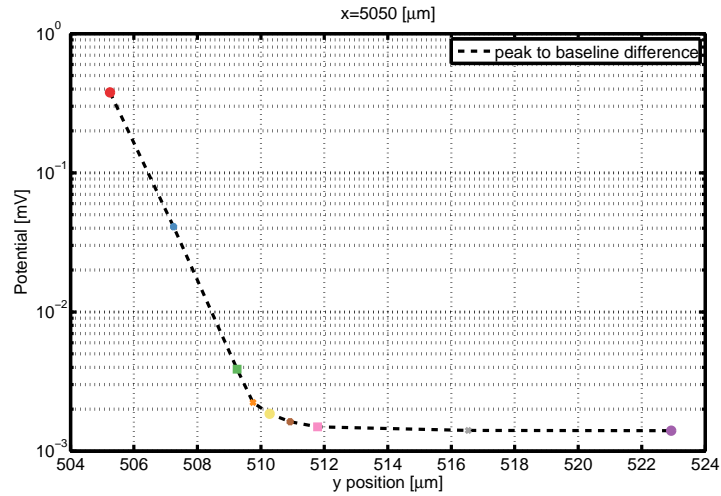


Figure 5.14: **Decay of peak-to-baseline in EAP over Debye layer.** For the EAPs in fig. 5.12, the difference between peak and baseline potential (value at $t = 2$ ms) is plotted on a log-scale, showing an abrupt switch in the fall-off rate.

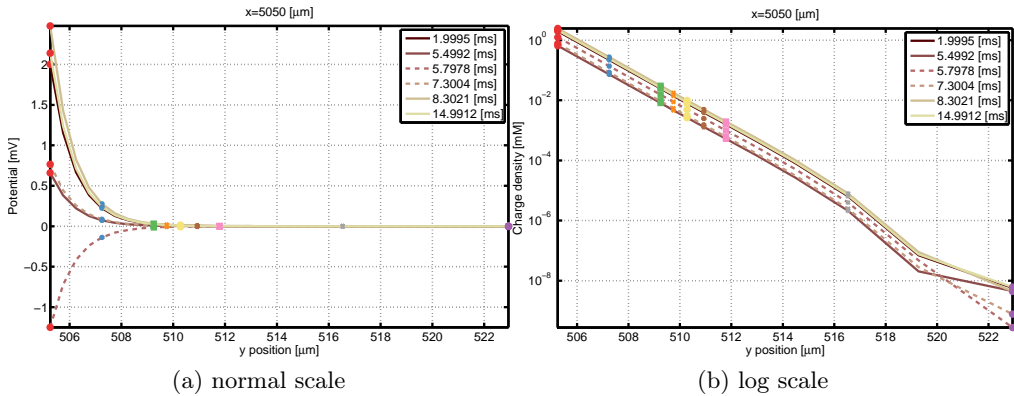


Figure 5.15: **Charge density profile over Debye layer at various time points.** As in fig. 5.13, the left panel is in normal scale, the right panel are the log-scaled absolute values. The kinks between $y = 518$ nm and $y = 520$ nm are a consequence of the charge density switching signs, as it is already close to zero. In both panels, the plotted markers correspond to the positions of the time courses in fig. 5.12.

the dominating AP echo. In this small range close to the membrane, the PNP system seems to be described very well by the stationary Poisson-Boltzmann eq. (2.22), letting the concentrations undergo an exponential decay towards their bulk values. As the right-hand side of eq. (2.22) is an exponential function, the potential – after integrating twice – must also follow an exponential function and decay rapidly towards the “bulk potential”, which is zero for an electroneutral bulk solution. However, this bulk potential is never reached, as by traversing through the Debye layer, the influence of membrane fluxes increases, and the Poisson-Boltzmann equation does not provide a suitable representation anymore.

5.2.2.6 Extracellular Concentrations

In analog to the nearfield and Debye layer potential in figs. 5.9 and 5.11, we can also look at concentration snapshots, see fig. 5.16. We stated above that the charge density (fig. 5.16d) is always oriented opposed to the potential, as counterions are attracted to and co-ions repelled from the membrane, consistent with the Poisson-Boltzmann eq. (2.22). In fig. 5.15b, we also saw that the charge density drops exponentially over the Debye layer at any given time, satisfying the equilibrium Boltzmann distribution on the right-hand side of eq. (2.22). This indicates that in the Debye layer, the stationary Poisson-Boltzmann equation is valid also for the timescales present in this instationary case. The membrane transshipment happens so fast it can be considered instantaneous.

As expected from the charge density profile in fig. 5.15, the concentrations fall off very quickly. Close to the membrane, sodium (fig. 5.16a) as the main counterion (at equilibrium, the membrane is negatively charged from the extracellular point of view) shows an inverted AP shape, since it is repelled by the incoming (positive) action potential. Chloride as a co-ion, on the other hand, follows the AP shape due to its negative charge (fig. 5.16c). A slightly different picture shows for potassium in fig. 5.16b, whose shape shows similarities to the sodium profile, albeit somewhat distorted. Since the extracellular potassium bulk concentration is two orders of magnitude lower than those of sodium and chloride, the previous reasoning of membrane currents being hidden in the steep concentrations gradients does not fully apply here. In contrast to the other two ion species, a small variation by membrane currents in absolute numbers has a larger effect due to its lower bulk concentration, so that the influence of membrane currents can be clearly seen in the potassium concentrations, especially the increase after opening of K_v channels to the left of the profile. The total charge density in fig. 5.16d follows the counterions, mainly the higher concentrated sodium ions.

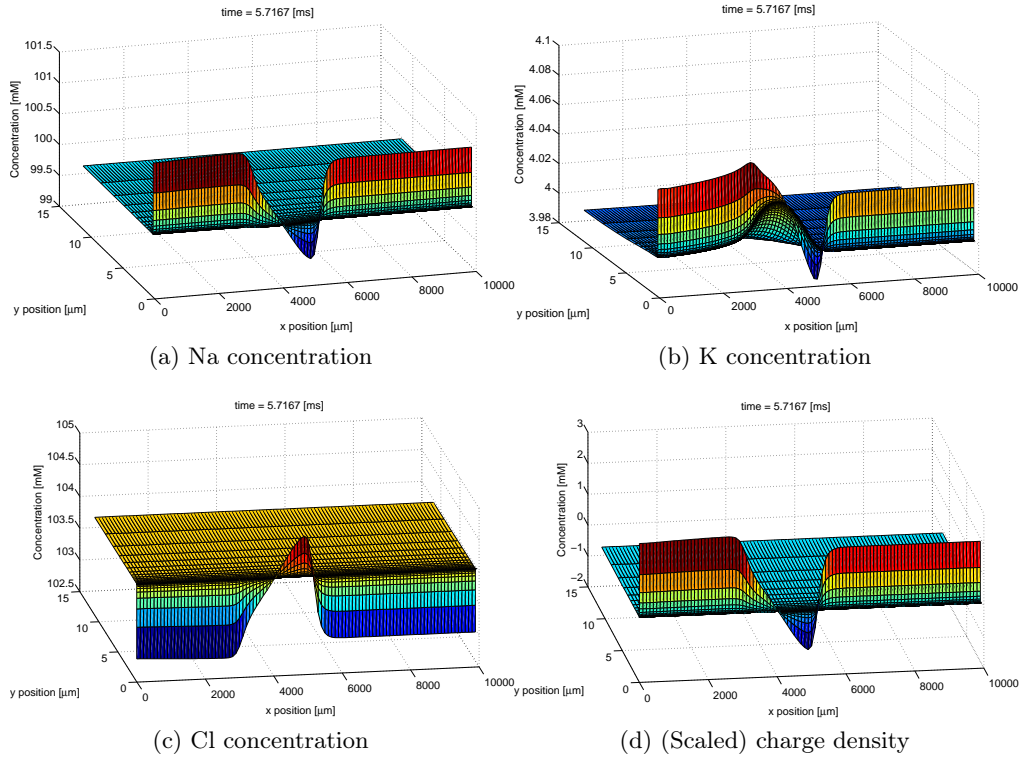


Figure 5.16: **Snapshot of the Debye layer and nearfield concentration profiles.** The concentration profiles of all three ion species and the charge density are plotted up to a distance of about $10\mu\text{m}$ from the membrane. The strong concentration gradients close to the membrane demonstrate the influence of the Debye layer. The charge density can be seen to be dominated by the sodium concentration.

5.3 Comparison with Other Models

In order to compare the electrodiffusion results with existing models for the extracellular potential, a 10-fold finer x -resolution of $h_x = 10 \mu\text{m}$ was used to ensure an accurate resolution of membrane dynamics, especially membrane potential and channel currents. A finer y -resolution of $h_y^{\text{max}} = 10 \mu\text{m}$ did not yield a notable difference, so it was kept at $h_y^{\text{max}} = 100 \mu\text{m}$.

5.3.1 Line Source Approximation

The **LSA** model as an analytical solution of eq. (2.38) on page 37 has been introduced in section 2.3.5.2. It can be used to calculate the extracellular potential as a superposition of current contributions of a finite number of line segments, as produced e.g. by a NEURON simulation. But also the extracellular field computed by the electrodiffusion model can easily be compared with the **LSA** results, as the membrane currents can be calculated directly as the sum of ionic currents – known from the membrane flux boundary conditions – and a capacitive current²

$$I_j = I_{\text{ionic}}(x) + I_C(x) \quad (5.6)$$

$$= eN_A \left(\sum_{i=1}^N f_i^{\text{memb}}(x) + f_C(x) \right), \quad (5.7)$$

where the capacitive flux f_C was calculated as described in section 5.2.2.2, and x is the center of the membrane-extracellular interface corresponding to line j in **LSA** notation.

The resistivity ρ was chosen manually in such a way that the positive peaks P1 of electrodiffusion and **LSA** approximately matched for moderately large distances from the membrane ($> 5 \mu\text{m}$ in this case). This resulted in a value of $\rho = 72 \Omega \text{cm}$, corresponding to a conductivity of $\sigma = 1.39 \text{S/m}$, which is in good agreement with [16] and 20-27.5% lower than in [42]. This is plausible, respecting that measurements in [16, 42] were carried out at room/body temperature, while the data in this work was generated at a lower temperature of $T = 6.3 \text{°C}$, causing a lower conductivity. It should also be noted that this value matches well with the conductivity of $60 \Omega \text{cm}$ – $70 \Omega \text{cm}$ for *Ringer's fluid*, an electrolytic solution that is isotonic to body fluids like blood or cerebrospinal fluid [40].

²Note that unfortunately, the capacitive current I_C was omitted in [82]. A correction is available under <http://dx.doi.org/10.1016/j.bpj.2013.12.026>. This work already contains the corrected version.

Figure 5.17 shows the comparison of potential time courses for *LSA* and *electrodifusion (ED)* models at various distances from the membrane. Additionally, the difference between the two curves is plotted.

One can see that both signals agree very well for larger distances from the membrane ($> 5 \mu\text{m}$). For distances below a few microns, however, the deviations are clearly visible. Especially the latter part of the signal after N1 shows qualitative differences. A special case is the potential at a point directly adjacent to the membrane in fig. 5.17a, which is exclusively dominated by the *AP* echo in the electrodiffusion simulation, while *LSA* predicts a much smaller amplitude. Continuing across the Debye layer and into the nearfield regime, both signals show better matches for increasing distances, where the *EAP* calculated by the electrodiffusion model tends to have smaller amplitudes and a tail (P2 – S – P3) that is shifted downwards. A notable difference is also the potential answer to the stimulus artifact, which happens instantaneously in the form of a rectangle pulse in *LSA*, while the gradual electrodiffusion answer is much smoother in comparison.

5.3.1.1 Separate Inspection of EAP Contributions

To further investigate the reason for the nearfield differences between *ED* and *LSA* results, an additional simulation was carried out to eliminate the contribution of ionic currents. To this end, an “echo simulation” was set up with the same parameters as before, but with all membrane channels closed, resulting in zero membrane fluxes. The lower potential Neumann-0 boundary condition was replaced by a time-dependent Dirichlet boundary, generated separately by an action potential simulation as before (the “source simulation”). This resulted in a setup where the contributions of ionic membrane currents are eliminated, enabling the assessment of the membrane potential and induced capacitive currents alone. This can, in turn, be compared to the *LSA* model with only capacitive currents, allowing for the evaluation of differences of capacitive contributions separately.

A notable difference between these simulations and the one considered before is the usage of discrete Dirichlet values from the source simulation as a boundary condition in the passive membrane solution. An interpolation of these discrete values generated kinks in the lower potential boundary condition, which made the system very susceptible to numerical oscillations. The kinks got picked up and amplified quickly in the numerical solution, especially at larger distances from the membrane, probably due to the threshold volume scaling. When using a finer space grid – as mentioned above, with mesh parameters $h_y^{\text{max}} = 100 \mu\text{m}$ and $h_x^{\text{max}} = 10 \mu\text{m}$ – together with a finer time grid, with upper time step values

5.3 Comparison with Other Models

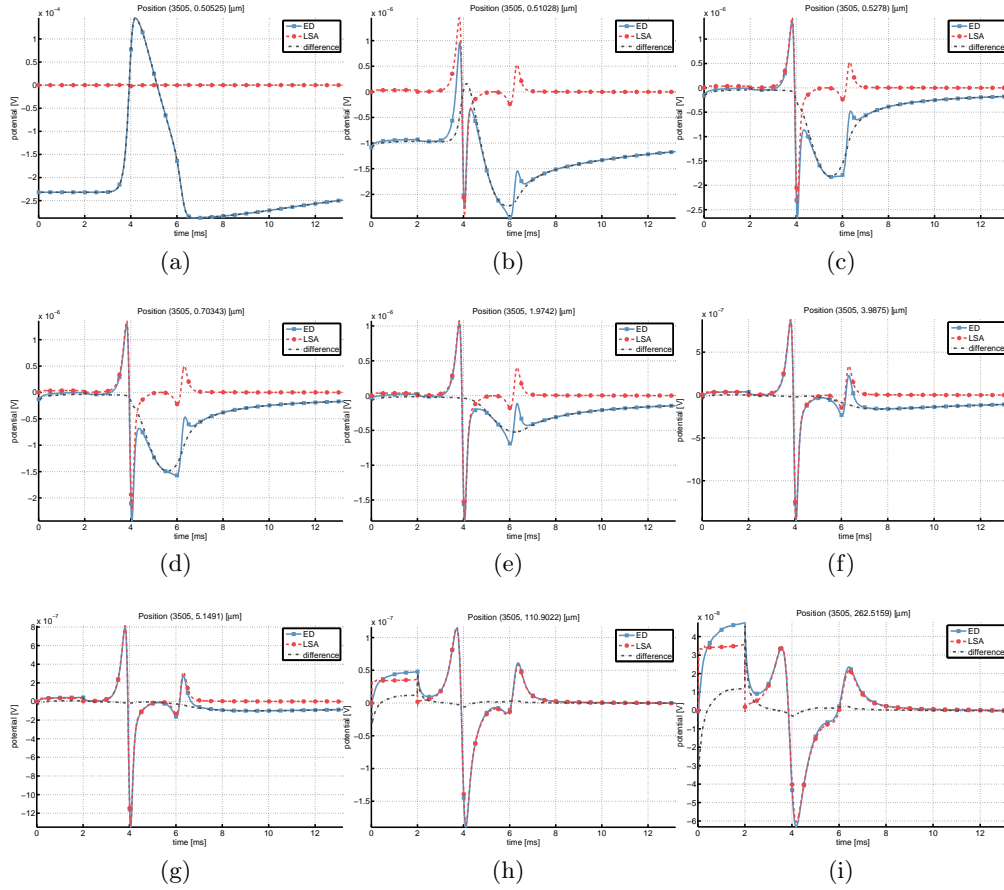


Figure 5.17: **Comparison of ED and LSA potentials.** The time courses of extracellular potentials calculated by electrodiffusion (**ED**, *solid lines*) and line source approximation (**LSA**, *dashed lines*) models are compared at different distances from the membrane (a-i) for a fixed x -coordinate. Additionally, the difference (*dash-dotted lines*) between the two curves is shown. The different behavior for the part between N1 and P3 is apparent at lower distances, but for larger distances, a good agreement can be found.

of $\Delta t_{\max} = 10 \mu\text{s}$ and $\Delta t_{\max, \text{AP}} = 1 \mu\text{s}$ for the source simulation, and additionally using the same space and time grid in the passive membrane setup, the oscillations could be successfully eliminated.

Figure 5.18 shows the EAP time courses at the same points as before for the full ED simulation in fig. 5.8, in comparison with LSA calculated using only the capacitive flux $I = I_C$. As expected, Debye layer potentials deviate completely, as the AP echo simulation still captures strong Debye layer gradients. But interestingly, the match outside the Debye layer is remarkably well. Notable differences can be made out during the on- and offset of the stimulation artifact until $t = 2 \text{ ms}$.

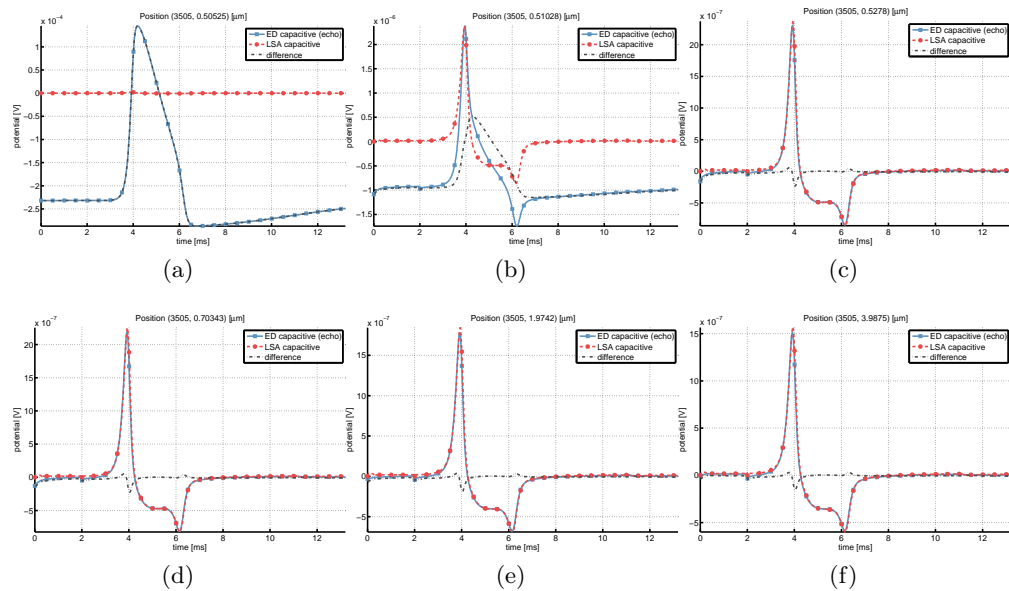


Figure 5.18: **Comparison of the capacitive component of the EAP for ED and LSA.** This shows only the capacitive component of the full EAP in fig. 5.17 for LSA (*dashed lines*) and ED (*solid lines*) model, as generated by the echo simulation.

Since now the potential from the source simulation as well as the reduced AP echo simulation are available at the same points in space and time, it is possible to simply subtract the potentials of both electrodiffusion simulations from each other, yielding a potential due to ionic currents only. In analogy to the isolated capacitive component in fig. 5.18, this allows us to assess the isolated

ionic component of the **EAP** in fig. 5.19. In contrast to the capacitive component, the potential due to ionic currents shows bold deviations also for large distances. It now becomes clear that the nearfield deviations between electrodiffusion and **LSA** models can be tracked down to the ionic component of the **EAP**.

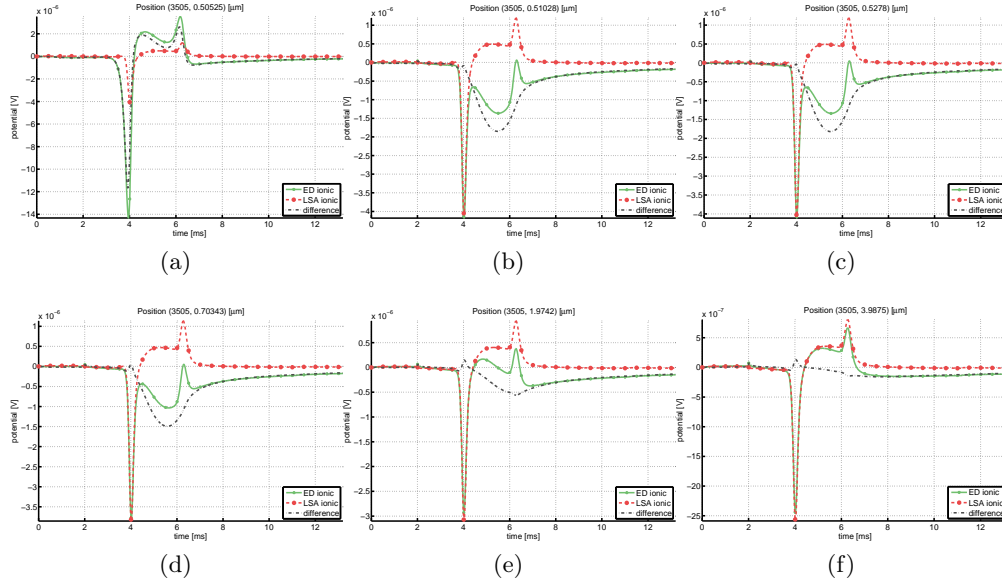


Figure 5.19: **Comparison of the ionic component of the EAP for ED and LSA.** This shows only the ionic component of the full **EAP** in fig. 5.17 for **LSA** (dashed lines) and **ED** (solid lines) model, as calculated by subtracting results of the echo simulation from the source simulation.

We have now arrived at a point where we can try to explain the difference between the full electrodiffusion model and volume conductor models like **LSA**. The deviations have been found to originate in the ionic part of the extracellular potential, the main contributor. The literature reveals several hints for an explanation: in [41], the authors describe a region of potassium accumulation around a nerve fiber with an approximate size of 30 nm, which is referred to as a *Hodgkin-Frankenhaeuser space* or simply *Frankenhaeuser space* in later publications. This might actually be a different name for the *diffuse layer* from the **EDL** field, see section 2.3.2. In his analysis of the **PNP** equations, Mori [73] describes the existence of an intermediate *diffusion layer* of size $\sqrt{\gamma D}$ between Debye layer and bulk solution and also mentions Frankenhaeuser spaces in this

context. However, the dimension of his diffusive layer is orders of magnitude larger (about $30\ \mu\text{m}$ for a Debye length of $\gamma_D = 0.9\ \text{nm}$) than the dimension mentioned in [41]. This difference in magnitude does not have to be contradictory: while the diffuse layer is the region where concentrations deviate from their bulk values, Mori's diffusion layer describes the region in which these concentration deviations have an influence on the potential, such that the electroneutrality assumption can not be used. In any case, the existence of such a layer and its influence on the EAP might be the very difference we see in fig. 5.19 and therefore the one that is not captured in volume conductor models.

5.3.2 Numerical Solution of the Volume Conductor Equation

In addition to the special case of the line source approximation, the underlying PDE was also solved numerically, to which we will refer as VC in the rest of this section, in dissociation to the analytical LSA. Our main goal was to have a second model to compare with, and to ensure our numerical methods work correctly, as in this case VC should give the same results as LSA. The same computational grid and exterior boundary conditions as for the source simulation from the previous section were used for the VC simulation, the only difference being that we only used the extracellular domain Ω_{ES} of the source grid for faster computations, eliminating the cytosol unknowns from the system. The internal membrane flux conditions were loaded from a previous PNP simulation. Large parts of the implementation infrastructure described in section 4.2.2 could be reused for this, with the only major change being the usage of a different parameter class `LaplaceParameters` for the operator `Dune::PDELab::ConvectionDiffusionFEM`.

To validate the numerical results, they were first compared to the LSA model, which should give an exact match within the tolerance of numerical and boundary errors present in the numerical algorithm. Figure 5.20 shows this comparison. As before, a wide range of the extracellular space was chosen, including very small distances from the membrane. This serves to detect deviations close to the membrane current source, because the LSA uses a collapsed line source, while the numerical simulation uses the full cylinder surface source. No notable differences can be found even very close to the membrane, suggesting that the representation as a line source is a valid approximation. At larger distances from the membrane, the artifact of the grounding Dirichlet boundary in VC becomes visible. Since this error is an absolute one (cf. the elaboration in section 3.3), its relative influence grows with increasing distance from the membrane. Within the considered range, i.e. at sufficient distances to the upper boundary, the error is still acceptable.

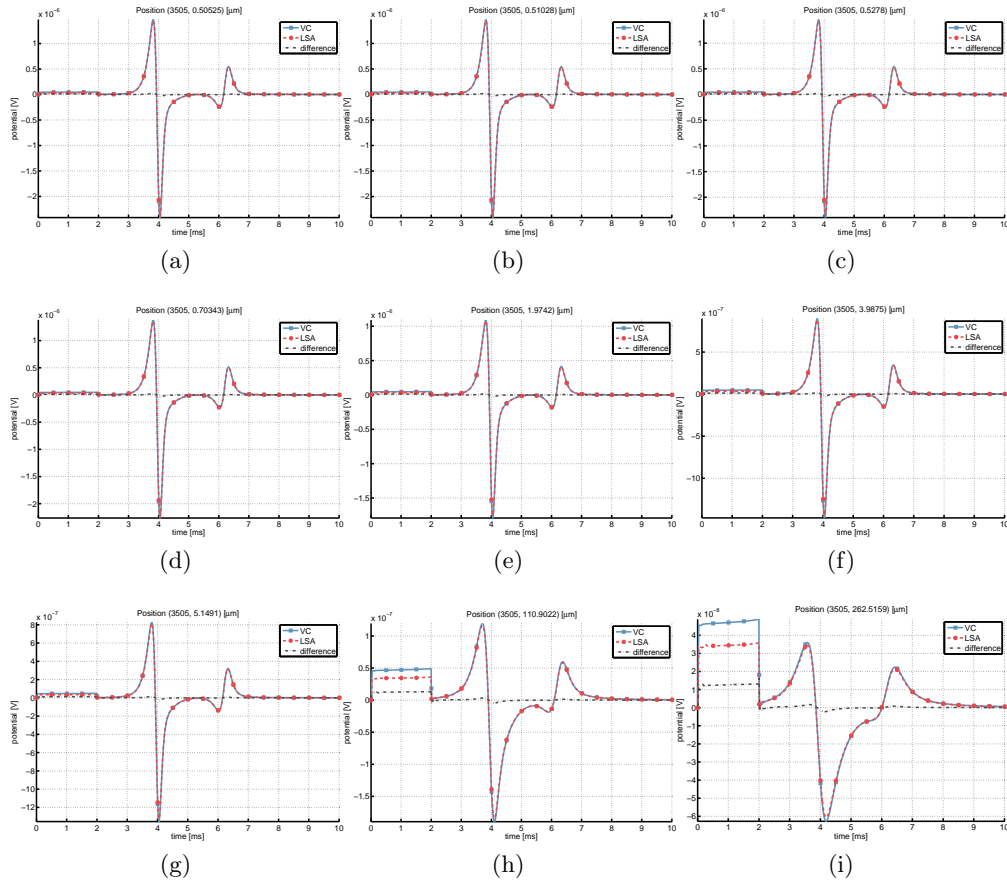


Figure 5.20: **Comparison of numerical and LSA solution to the volume conductor equation.** A wide range of extracellular points was used to assess the accuracy of the numerical solution (*solid lines*). The match is very good even at small membrane distances, indicating that the line source approximation (*dashed lines*) provides a suitable reduction of the full cylinder geometry. At larger distances, the influence of the upper Dirichlet boundary leads to notable deviations from *LSA*.

After ascertaining the numerical solution is correct, it can be compared to the electrodiffusion results generated on the same computational grid, shown in fig. 5.21. Unsurprisingly, the plot shows the same situation as in fig. 5.17.

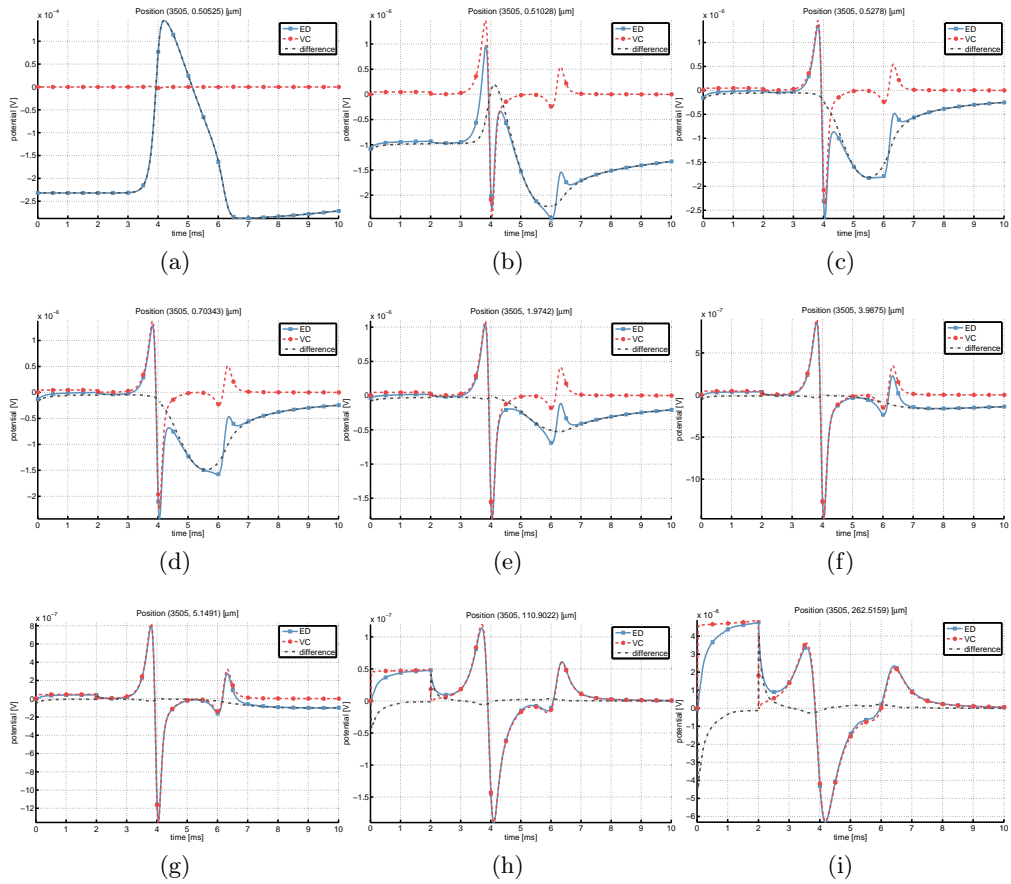


Figure 5.21: **Comparison of ED and VC results.** Both numerical simulations (ED, solid lines; VC, dashed lines) were generated on an identical computational grid and compared at the same positions as before in fig. 5.17, showing essentially same differences (dash-dotted lines).

5.4 Summary

The modeling approach of chapter 3 was applied to an axon with homogeneous channel densities. The resulting extracellular potentials can largely be attributed to the membrane currents following the HH system, consistent with existing models based on volume conductor theory.

However, close to the membrane, where the Debye layer was explicitly resolved, we found significant differences, as large concentration and potential gradients towards the membrane dominate the membrane currents. In accordance with the Poisson-Boltzmann eq. (2.22), the concentrations are screened exponentially, relaxing to their bulk values over the extent of the Debye layer. The changes in concentration triggered by membrane fluxes here are small compared to the deviations of concentrations from their bulk values in the Debye layer, therefore this does not have a significant effect on the potential. While this region with prominent characteristics is only about 10 nm wide, its effect on the EAP shape is clearly visible up to distances of at least 5 μm , underlining the importance of this modeling approach.

We note that this is consistent with the PNP system analysis in [73], where a diffusion layer with an extent on the order of $\sqrt{\gamma D}$ is found. We attribute the deviations between LSA and electrodiffusion model to the presence of such a layer with associated concentration redistributions and its effects on the potential. On the contrary, the good agreement with the LSA model in large parts of the domain can be regarded as a validation of our model. The only free parameter in the LSA model, the resistivity ρ , was fitted to yield the optimal agreement with our data, and turned out to match very well with reported values from the literature.

From the computational point of view, even this seemingly simple example of a homogeneous axon imposes a severe challenge on the numerical solution. At the membrane, sharp discontinuities (in concentrations) as well as steep gradients (concentrations and potential) occur, which were accounted for by a sub-nanometer grid resolution in radial direction. In longitudinal direction and off the membrane, large mesh sizes on the order of 10^2 – 10^3 μm were used. The usage of appropriate linear solvers was crucial to cope with the introduced grid anisotropy. The multi-scale character of the system also shows in the generated solution. For example, the amplitudes of the potential differ by several orders of magnitude between the intracellular values (about 100 mV), the extracellular values close to the membrane (up to a few 100 μV in the Debye layer), and some microns away from the membrane (fractions of 1 μV).

6 Model of a Myelinated Axon

If our brains were simple enough for us to understand them, we'd be so simple that we couldn't.

(Ian Stewart)

To handle myelinated axons, some major changes are required in the model, causing further adjustments in the discretization and the numerical algorithms. These changes will be presented in this chapter as well as their influence on the simulation results.

6.1 Representation of Myelin in the Model

The relevant changes to include a myelin sheath around the axon – with respect to the general model from chapter 3 – concerns the membrane. Following the model assumption of a cylindrical axon, the myelin wrapper can be seen as a hollow cylinder that encloses the axon, albeit with a larger radius than the axonal membrane.

In reality, myelin is not a homogeneous material, it rather consists of multiple repeating layers of lipid – consisting of polar heads and hydrophobic tails – and water (cf. [54, chapter 6.17]). To simplify the myelin representation, the presence of these multiple layers with varying dielectric constants is neglected. Instead, it is assumed to be a homogeneous material with an effective electric permittivity ϵ_{myel} , which does not impose a problem if one is not interested in the fine-granular potential profile across the myelin sheath, but rather the effective membrane potential.

Estimated values for ϵ_{myel} are within the range of 3–8.5 [54, 78]. An increased myelin sheath thickness or tighter binding with the membrane might, however, result in a decreased value [72]. Furthermore, the effective value should also respect the lower axonal membrane permittivity of $\epsilon_{\text{memb}} = 2$, which is located below the myelin sheath. For the rest of this chapter, we made the more or less arbitrary choice of $\epsilon_{\text{myel}} = 6$, which can easily be changed in the application's config file.

The computational domain from section 3.2 needs to be adjusted to represent the myelin sheath as well, as shown in fig. 6.1, where the myelin is included explicitly. It can be seen that the dashed membrane interface is not a straight line anymore, it rather consists of multiple horizontal and vertical parts. If the Debye layer was to be resolved for this membrane interface shape, as suggested in fig. 6.1, the number of grid points would become very large, resulting in an intractable number of DOFs.

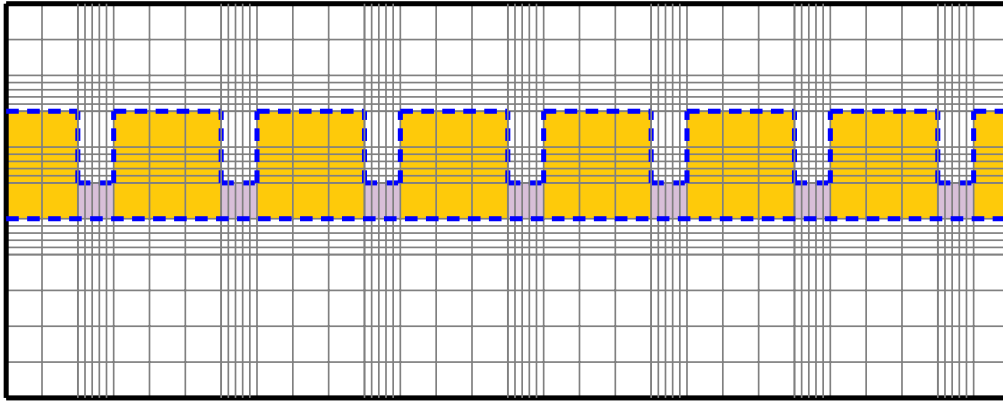


Figure 6.1: **Computational domain with explicit myelin representation.** Due to the varying membrane thicknesses, the resolution of the Debye layer would require an unnecessarily high number of unknowns when using a tensorial grid.

Fortunately, a property of the potential can be utilized. It can be observed that, in a first approximation, the potential shows a linear decay across the membrane. This is due to the fact that the Debye length γ_D is much smaller than the membrane thickness d_{memb} , meaning that charges on both sides of the membrane only have a limited effect on the electric field inside the membrane. Arguing mathematically, the right-hand side of the Poisson eq. (2.43) is close to zero, thus yielding a potential with zero curvature.

This was validated in simulations with a fine membrane resolution ($n = 10$ membrane elements in y -direction) and compared to the case $n = 1$, showing a negligible deviation in fig. 6.2. As described in section 4.2.2, this also enables blocking the system matrix, which is of great advantage for the performance of preconditioner and linear solver.

When assuming a linear potential course over the membrane, a nice property can be utilized: a membrane with thickness d_1 and permittivity ϵ_1 as in fig. 6.3a

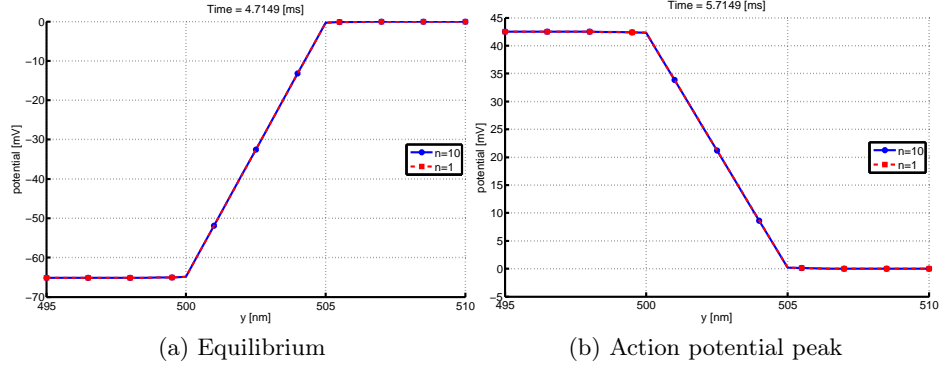


Figure 6.2: **Comparison of potential profiles for different membrane resolutions.** Potentials across the membrane at a fixed x -position of 5 mm (center of the axon) at two exemplary time points: resting state at $t = 4.7$ ms (*left*) and AP peak at $t = 5.7$ ms (*right*). Solid lines show the profile for $n = 10$ membrane elements and dashed lines for a single membrane element in y -direction. Differences are negligible at every point in time (not shown).

will have the same membrane potential decay as another membrane with thickness $d_2 = \frac{d_1}{2}$ and permittivity $\epsilon_2 = \frac{\epsilon_1}{2}$ in fig. 6.3b. With this, the myelin parts of the membrane from fig. 6.1 can be “collapsed” onto the axonal membrane by changing the permittivity accordingly, see fig. 6.4. This results conceptually in the same computational grid as for an unmyelinated axon, with only one finely resolved Debye layer on each side of the membrane.

Following the previous explication, the collapsed myelin permittivity ϵ'_{myel} can be calculated as

$$\epsilon'_{\text{myel}} = \epsilon_{\text{myel}} \frac{d_{\text{memb}}}{d_{\text{myel}}} ,$$

with the (nodal) membrane thickness d_{memb} as before and the myelin thickness d_{myel} , which was chosen to be 500 nm in the following. With this, we arrive at $\epsilon'_{\text{myel}} = 0.06$.

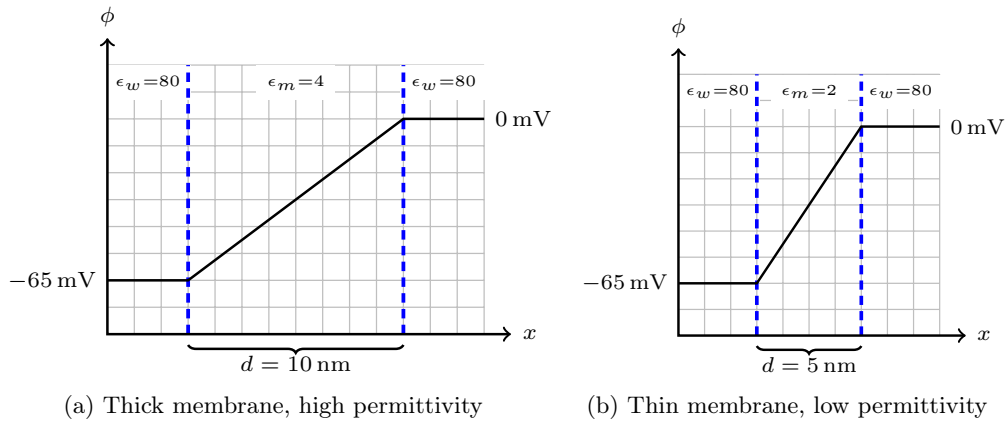


Figure 6.3: **Relation of linear potential decay and membrane thickness/permittivity.** Doubling membrane thickness and permittivity (*left*) results in the same membrane potential difference when the potential is linear across the membrane.

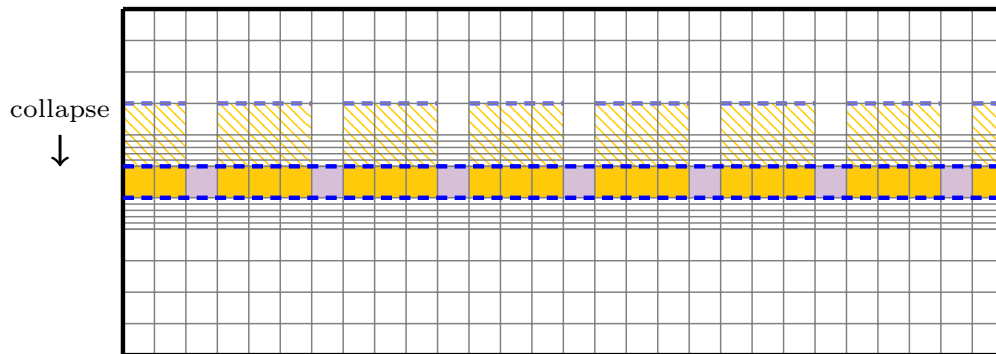


Figure 6.4: **Computational domain with collapsed myelin representation.** Myelination is represented implicitly, by collapsing the membrane to a homogeneous thickness and instead using effective permittivities that compensate for the thickness modification.

6.2 Simulation Setup

The changes in the axonal geometry described above require a number of changes in the numerical implementation, which will be presented in the following.

6.2.1 Modified Grid Generation

The grid spacing in x -direction can not be equidistant as before, since the different material properties of myelin and nodes of Ranvier have to be accurately represented without using a prohibitively small grid size. A related problem arises through the jumping material coefficients, necessitating the introduction of fine transition intervals between myelin and nodes of Ranvier. Inside the membrane groups – where the permittivity is constant – a coarser, equidistant mesh size may be used. Analogous to the y -direction, abrupt changes in mesh width are avoided.

This results in a mesh generation procedure as follows: the transition intervals between myelin and node of Ranvier – as well as the node of Ranvier itself – are finely resolved with a minimum mesh width of h_x^{\min} . This value should be a fraction of the node width $l_{\text{node}} = 1 \mu\text{m}$, here we used $h_x^{\min} = 100 \text{ nm}$. Outside the transition interval, the grid spacing is smoothly increased following a geometric series up to a maximum equidistant grid size of $h_x^{\max} = 10 \mu\text{m}$ for the myelin parts. This procedure is mirrored towards the next node of Ranvier and repeated for each node until the end of the x -interval. The length of an internode segment $l_{\text{myel}} = 999 \mu\text{m}$ was chosen such that the starting coordinates of two successive nodes of Ranvier are always 1 mm apart.

For an axon with 10 nodes of Ranvier and above parameters l_{node} , h_x^{\min} and h_x^{\max} , this will increase the overall number of grid points by a factor of 7-8 in comparison to the unmyelinated case. This number is acceptable, especially when running the simulation in parallel, where even a much finer grid is still tractable.

In the following, we will consider a reference setup with a myelinated axon of length 48 mm consisting of 48 nodes of Ranvier. The number 48 derives from the fact that the simulation was run on a machine with 48 cores, allowing an easy partitioning that puts one node of Ranvier on each processor, see below. All relevant parameters are listed in table 6.1.

Table 6.1: Simulation parameters for the myelinated axon model.

Parameter	Value	Unit	Description
GRID			
x^{\max}	48	mm	Domain size (x -direction)
y^{\max}	10	mm	Domain size (y -direction)
y_{memb}	500	nm	Radius of the axon
d_{memb}	5	nm	Membrane thickness
d_{myel}	500	nm	(Implicit) myelin thickness
l_{node}	1	μm	Width of one node of Ranvier
l_{myel}	999	μm	Width of myelin internode
h_x^{\min}	100	nm	Minimum mesh size in x -direction (node of Ranvier)
h_x^{\max}	10	μm	Maximum mesh size in x -direction (myelin)
h_y^{\min}	0.5	nm	Minimum mesh size in y -direction (Debye layer)
h_y^{\max}	100	μm	Maximum mesh size in y -direction
#DOFs	5 250 448	1	Total number of unknowns
PHYSICS			
ϵ_{memb}	2	1	Membrane permittivity (node of Ranvier)
ϵ'_{myel}	0.06	1	Collapsed membrane permittivity (myelin)
ϵ_{elec}	80	1	Electrolyte permittivity
$[\text{Na}]_0^{\text{CY}}$	12	mM	Intracellular Na^+ bulk concentration
$[\text{K}]_0^{\text{CY}}$	125	mM	Intracellular K^+ bulk concentration
$[\text{Cl}]_0^{\text{CY}}$	137	mM	Intracellular Cl^- bulk concentration
$[\text{Na}]_0^{\text{ES}}$	100	mM	Extracellular Na^+ bulk concentration
$[\text{K}]_0^{\text{ES}}$	4	mM	Extracellular K^+ bulk concentration
$[\text{Cl}]_0^{\text{ES}}$	104	mM	Extracellular Cl^- bulk concentration
D_{Na}	1.33×10^{-9}	m^2/s	Na^+ diffusivity
D_{K}	1.96×10^{-9}	m^2/s	K^+ diffusivity
D_{Cl}	2.03×10^{-9}	m^2/s	Cl^- diffusivity
g_{Na_v}	1200	mS/cm^2	Conductance of the voltage-gated Na channel

Parameter	Value	Unit	Description
PHYSICS			
g_{K_v}	360	mS/cm ²	Conductance of the voltage-gated K channel
g_L	0.5	mS/cm ²	Total leak conductance
NUMERICS			
reduction	5×10^{-6}	1	Newton reduction
absLimit	1×10^2	1	Newton absolute limit
t_{end}	20	ms	Simulated time
Δt_{min}	0.05	μs	Minimum time step
Δt_{max}	10	μs	Maximum time step
$\Delta t_{\text{max,AP}}$	10	μs	Maximum time step during AP
it_{min}	10	1	Newton iteration threshold for time step increase
it_{max}	30	1	Newton iteration threshold for time step decrease

6.2.2 Permittivity Smoothing

Making sure the grid resolves the membrane group transitions is crucial, but the membrane permittivities will still exhibit a sharp discontinuity at these transitions. Therefore, membrane permittivities are smoothed at the start of a simulation by applying the `smoothstep` function

$$\text{smoothstep}(x) = 3x^2 - 2x^3 \quad (6.1)$$

to transitions intervals, effectively applying cubic Hermite interpolation. The extent d_T of a transition interval depends on the width of the nodes of Ranvier. Here, it was chosen to be 100 nm, i.e. $\frac{1}{10}$ of the node width. This results in the membrane permittivity profile in fig. 6.5; the effects of the smoothing operator can clearly be seen when zooming into a node transition interval in fig. 6.5b. This prevents numerical oscillations at sharp coefficient discontinuities, to which the standard finite element method is very susceptible.

6.2.3 Parallelization and Adjustments of Numerical Methods

Due to the increasing computational demands by the inclusion of myelin into the model, the numerical solution was only run in parallel. As before, the grid

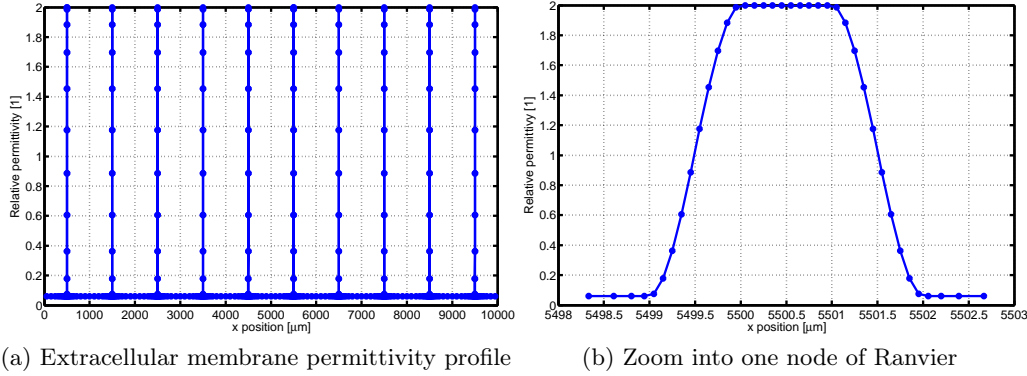


Figure 6.5: **Equilibrium permittivity profile.** The relative permittivities exhibit jumps at membrane group boundaries, resulting in a comb-shaped profile (*left*). The discontinuities were smoothed to prevent numerical oscillations, as can be seen in the zoomed-in transition region between myelin and node of Ranvier (*right*).

was partitioned only in x -direction. This not only ensures a joint treatment of membrane and adjacent electrolyte on one processor, it additionally avoids placing the highly dynamic nodes of Ranvier close to a processor boundary, which imposes a serious difficulty for the parallel solver. We ensured that nodes of Ranvier were always close to the middle of each processor subdomain, as illustrated in fig. 6.6.

The fact that the computational grid now has strongly varying mesh sizes in both x - and y -direction introduces severe problems for the linear solver. The `BiCGStab` iterative solver that was used for the unmyelinated case exhibited *breakdowns* when solving the problem in parallel, preferentially when using a higher number of processors. This is a known problem of the default `BiCGStab` algorithm [90], which can apparently be circumvented by so-called *look-ahead techniques* (see, e.g., [25]). These techniques have not yet been implemented in the `dune-istl` module.

But fortunately, there are a number of alternative linear solvers to choose from. For the problem at hand, we used a `GMRes` iterative solver, preconditioned by an `AMG` with an `ILU0` smoother and using SuperLU on the coarsest grid. The `GMRes` turned out to be very robust, albeit slightly less performant in comparison to `BiCGStab`. Using the `AMG` preconditioner in conjunction with an `ILU` smoother takes care of the strong grid anisotropy.

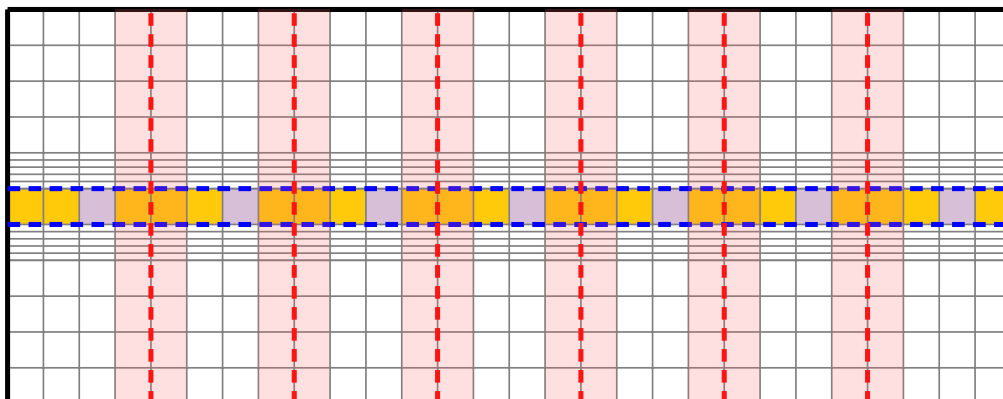


Figure 6.6: **Partition of the computational domain for the parallel case.** As in the unmyelinated case, the domain is divided by cuts in vertical direction. In the myelinated case, additional care has to be taken in order to not place the highly dynamic nodes of Ranvier close to a processor boundary, as suggested in the sketch.

6.2.4 Solver Scaling Tests

We again assessed the parallel performance of the numerical solution using the new linear solver backend. This time, however, we avoided strong scaling measurements, since this would inevitably place the processor boundaries close to nodes of Ranvier when choosing the number of nodes smaller than p . On the other hand, taking a setup with the number of nodes of Ranvier equal to the highest processor count $p = 48$ and using successively less processors to solve it was not an option either, since the sequential solution of such a large problem would have taken months.

Instead, a weak scaling test variant was used. The domain size was increased together with the number of processors, and the scalability S was calculated as

$$S(p_1, p_2) = \frac{t_1}{t_2}, \quad (6.2)$$

where we implicitly assume that the number of **DOFs** grows with a factor of $\frac{p_2}{p_1}$ and the number of unknowns is the same on all processors, although this is not exactly the case.

Results obtained in this manner for a fixed simulation time length t_{end} are, however, not comparable. Because the **AP** velocity is significantly higher com-

pared to the unmyelinated case, the action potential wave has already left the computational domain on the right boundary before it has exhibited a full depolarization–repolarization–hyperpolarization cycle at the left boundary, for small domain sizes. Only the largest simulation provides a sufficiently long axon of 48 mm to represent one full spatial wave length of the AP.

For smaller domain sizes, the AP rushes through the small extent of the axon in a fraction of the total simulated time; the longer the axon gets, the more time is spent by the traveling action potential within the computational domain. Consequently, the fraction of the simulation time that the axonal membrane is at rest is getting smaller, and the membrane is active for a larger fraction of time.

Of course, the active parts provide a much more difficult task for the numerical solution, as the system is more dynamic due to the large membrane currents, such that the solver needs more iterations and the total computation time increases. This is reflected in the initial defect (i.e., the norm of the residual $\|r^0\|$ at the beginning of each time step) in fig. 6.7, which shows an increase with the number of processors.

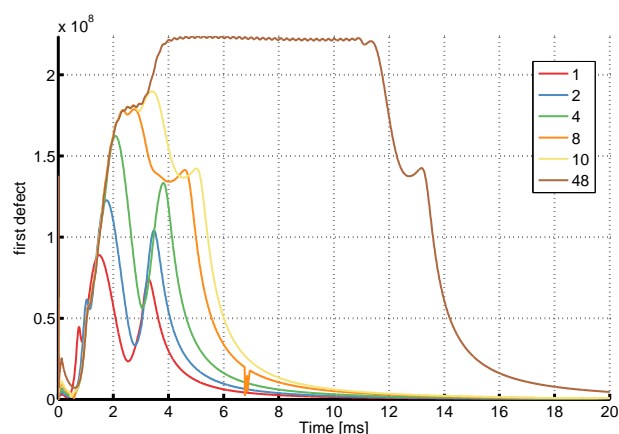


Figure 6.7: **Initial defect for different domain sizes.** The initial defect at the beginning of each time step is growing with processor count (and thus with domain size), since the AP wave spends a larger fraction of time within a longer axon.

Instead we consider, for each different p , only the time interval where the rightmost membrane element has a membrane potential of at least 60 mV. With this, we can calculate the average solver statistics only for the AP depolarization phase, shown in table 6.2. This evaluation presents a normalization excluding

the effects of different proportions of high or low system dynamics, as it only considers the time intervals of highest activity.

Table 6.2: **Solver statistics per time step for the parallel solution using different processor counts p .** All values are averages over all time steps. The total time includes matrix and residual assembly as well as the actual solution time by Newton’s method. The solver time includes both preconditioner setup and LS time, the LS time only the actual GMRes solve. The number of Newton iterations as well as the number of linear solver iterations per Newton iteration increase with problem size.

p	total time	S	assembler time [s]	solver time [s]	LS time [s]	LS it.	LS time / it. [s]	Newton it.
1	139.1566		119.2886	3.6972	1.134798	6.467105	0.186419	2.6746
2	147.3648	0.94	115.6824	15.4106	4.185112	21.094705	0.206684	2.7416
4	160.3079	0.87	119.7298	23.6005	5.913261	23.816343	0.252483	3.0371
8	265.5756	0.52	132.6605	113.0225	28.407855	64.485612	0.412704	3.502
10	246.884	0.56	128.8954	99.4551	27.090431	69.478478	0.368952	3.4438
48	199.0493	0.70	85.2555	99.7457	31.620872	90.697419	0.332404	3.111

Notable anomalies are the cases $p = 8$ and $p = 10$, which show inferior performance to the case $p = 48$. An inspection of the diagnostic output reveals that the time step size oscillates for these cases during the initiation of the AP, caused by a series of non-converging Newton iterations.

The overall solver timings normalized to the depolarization time interval show acceptable scaling properties, particularly for the case $p = 48$, which will be used to generate the simulation results considered in the following section.

6.3 Simulation Results

As before, equilibrium states and transient dynamics are considered separately. Most of the following results were obtained from the largest simulation with parameters according to table 6.1, but some plots were also obtained from a smaller, coarser setup with $x^{\max} = 10$ mm and $h_x^{\max} = 100$ μ m when looking

at point-data over time. In these cases, the domain extent did not make any difference, and deviations due to the x -resolution were negligible¹.

6.3.1 Some Notes on the Separate Visualization of Myelin and Node Values

Because of the comb-shaped permittivity profiles – as shown in fig. 6.5 – it is difficult to visualize the extracellular concentration and potential dynamics close to the membrane, as – to stay in the picture of a comb – the variation of these variables within “shaft” or “tooth” segments is much smaller than the difference of values across those two groups. This makes it impossible to see the variations within each group when plotting the complete domain.

Therefore, the vector of x -coordinates was partitioned into three disjoint sets: the set $X^{(\text{node})}$ of all node of Ranvier coordinates, the set $X^{(\text{tran})}$ containing the coordinates at transition intervals between myelin and nodes (specified by the maximum distance of d_T towards the nearest node coordinate), and the set $X^{(\text{myel})}$ containing the remaining x -coordinates of myelinated membrane cells, such that $x = X^{(\text{myel})} \cup X^{(\text{node})} \cup X^{(\text{tran})}$ holds. Note that this partition is done regardless of the y -coordinate, so the domain is cut into stripes parallel to the y -axis by this process.

With this partition of the x -axis, it is possible to visualize the values close to myelin and node of Ranvier parts separately by only using the data at x -coordinates given by $X^{(\text{myel})}$ or $X^{(\text{node})}$, respectively. We will use this later on to compare our results with the LSA model, for myelin and node stripes separately, and we will also take a close look at the highly dynamic transition interval between any of those two regions. The partition into three disjoint subsets is also important for the calculation of the equilibrium state from two different, small equilibrium simulations, as will be explained in the following.

6.3.2 Equilibrium States

As in the unmyelinated case, the equilibration simulations are performed on a grid with only one element in x -direction to speed up the procedure. Since we now have different membrane permittivities, the equilibrium concentration

¹The reason for using the smaller dataset when possible is a purely technical one: the size of the large dataset is about 419 GB on disk. In order to visualize this data, each variable has to be loaded separately for a domain subset and time subinterval of interest, as otherwise the data would not fit into the 8 GB of RAM on the author’s workstation. This results in tedious cycles of loading and clearing data for generating one single graph, which was much more comfortable with the smaller dataset of only 44 GB.

and potential profiles are quite different between myelin and nodes of Ranvier. Therefore, we need to carry out two different equilibration simulations, one for the myelin parts and one for nodes of Ranvier. At the beginning of the actual AP simulations, the values of both simulations are transferred to the fine grid, to the respective x -intervals of the domain associated with either myelin ($X^{(\text{myel})}$) or a node of Ranvier ($X^{(\text{node})}$). At transition intervals $x \in X^{(\text{tran})}$, the values of both equilibrium states are linearly interpolated with respect to the (smoothed) membrane permittivity at the current x -coordinate, calculated as described in section 6.2.2. This yields a smooth equilibrium state that is valid with respect to the membrane permittivity distribution on the fine grid. The system is then allowed to settle for a small number (< 20) of time steps before the stimulation electrode is switched on.

The equilibrium concentrations obtained hereby follow a distinct, comb-shaped distribution due to the variations of electrical permittivities along the axonal membrane, as depicted in fig. 6.8. The charge density largely follows the permittivity profile from fig. 6.5. The application of the smoothing operator to the permittivities turns out to be successful, as concentrations exhibit no oscillations, in contrast to the non-smoothed case (not shown).

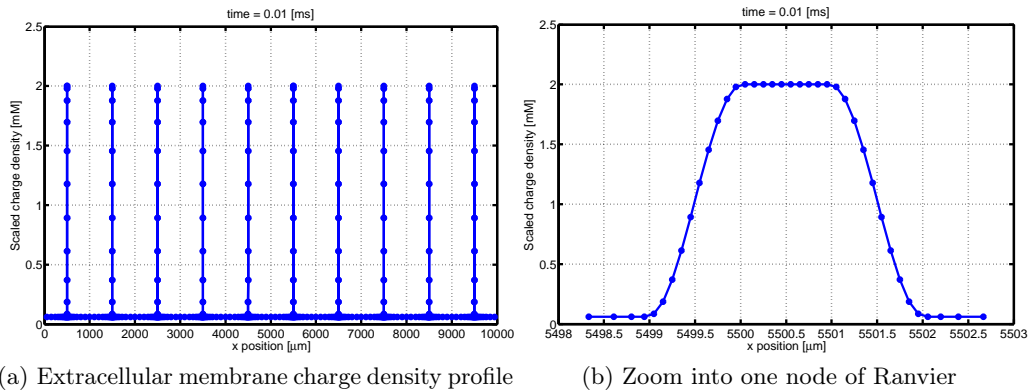


Figure 6.8: **Equilibrium charge density profile.** The scaled charge densities $\sum_i z_i n_i$ at equilibrium follow a distinct, comb-shaped distribution along the membrane (*left*). On the right, a zoom into one node of Ranvier reveals the jump in equilibrium concentrations due to different membrane permittivities at myelin and nodes of Ranvier.

6.3.3 Action Potential

As in the previous chapter, the stimulation electrode was mimicked by injecting a sodium rectangle pulse into the cell. The stimulation site was placed at the starting coordinate of the first node of Ranvier at $\mathbf{x}_{\text{stim}} = (500 \mu\text{m}, 0 \mu\text{m})$. This time, a drastically reduced stimulation current of about 10 pA over a duration of 3 ms was enough to elicit an action potential, as the isolation by myelin reduced the current leak over the membrane.

6.3.3.1 Intracellular Potential

In fig. 6.9, the generated action potential at different positions along the axon is shown on the right, yielding an impressive speedup of about a factor of 5. Another subtle difference is that the last curve shows a slightly higher amplitude than the other APs because the potential can not freely exit at the rear end of the axon (Neumann boundary for concentrations). Interestingly, this does not have an effect in the unmyelinated example, probably because the AP moves much slower, such that the excess in concentrations can drain off through membrane channels. In the myelinated example, the propagation is so fast that a swell of ions forms, showing a significant effect in the intracellular potential close to the right cell boundary. While this phenomenon clearly has to be regarded as a boundary artifact in this case, an accumulation of charge can be expected at axonal “dead ends” like synaptic boutons following an arriving action potential wave.

6.3.3.2 Membrane Flux

The ionic fluxes at a node of Ranvier in fig. 6.10a show the same shape as in the unmyelinated example from fig. 5.5 on page 80. This was expected, since the channel kinetics did not change. However, due to the factor of 10 in the channel densities, the amplitude is about a factor of 10 higher as well. This also explains why the capacitive flux is small against the others, resulting in a total membrane flux that is dominated by the ionic currents. A different picture shows at myelin parts, where no ion channels are present, resulting in zero ionic flux in fig. 6.10b. This causes the total membrane flux to be equal to the capacitive flux, which is smaller than at nodes of Ranvier due to the smaller permittivity (and therefore capacitance). The result is a markedly reduced amplitude of the flux in comparison with nodes of Ranvier.

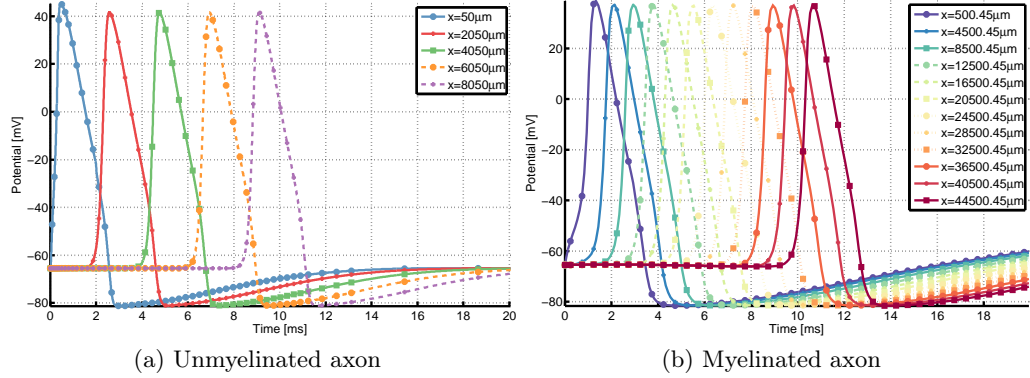


Figure 6.9: **Intracellular action potential propagation.** The action potential of the current setup (*right*, axon length 48 mm) is compared to the AP of unmyelinated axon from the previous chapter (*left*, length 10 mm). Please note the difference in spacings between measurement points. The increase in propagation speed is especially obvious, which is about a factor of 5 here.

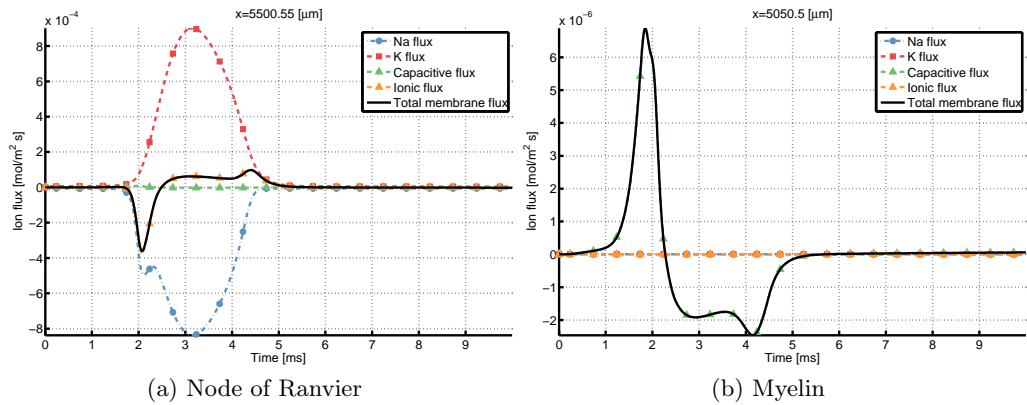


Figure 6.10: **Membrane flux at node of Ranvier and myelin.** The total membrane flux at a node of Ranvier (*left*) shows differences to that in the unmyelinated axon in fig. 5.5, as the capacitive component is very small. In contrast, at myelin parts (*right*), there are no ionic fluxes, so the capacitive flux dominates, although it has a markedly reduced amplitude in comparison to the node of Ranvier flux.

6.3.3.3 Debye Layer Concentrations

The extracellular concentration time courses in the Debye layer are shown in fig. 6.11. The delay in the activation of voltage-gated potassium (K_V) channels can be seen nicely in fig. 6.11b, showing first a negative deflection due to the arriving AP and then a larger positive peak as K_V channels open.

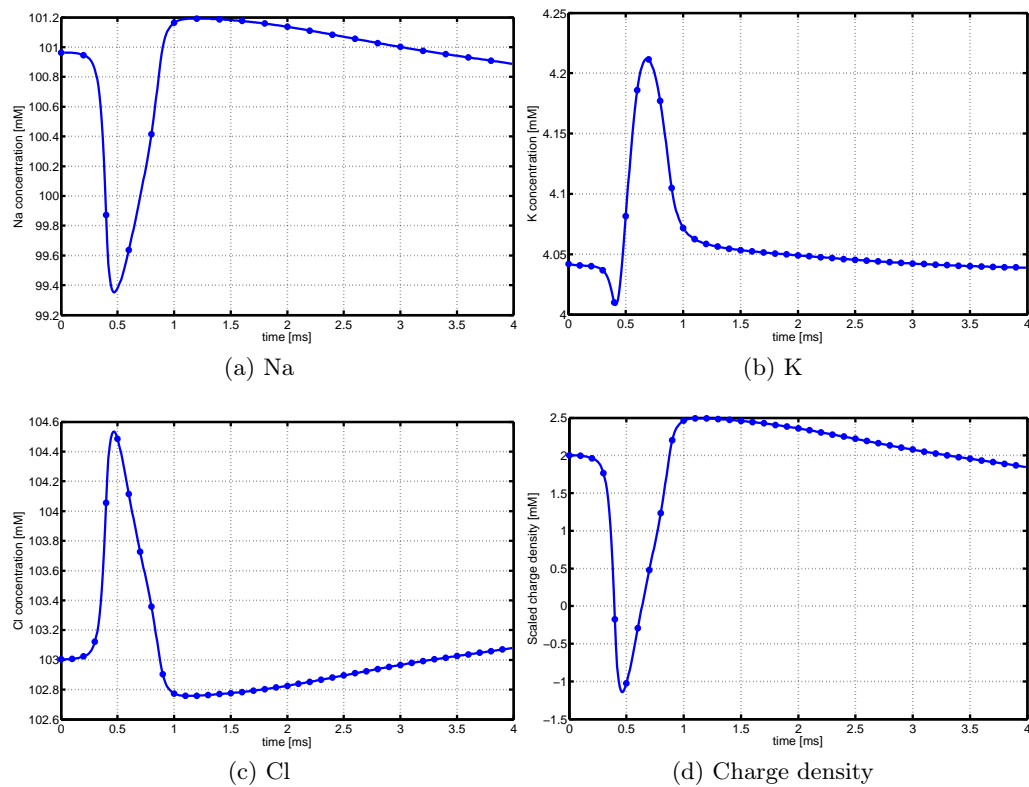


Figure 6.11: **Debye layer concentration time courses at node of Ranvier.** When compared with the concentration snapshots of the unmyelinated axon (cf. fig. 5.16), the shapes are quite similar. Sodium concentrations dominate the charge density and result in a chloride concentration with flipped sign. The potassium concentration shows a negative peak due to the arriving AP and a delayed positive response due to the activating voltage-gated K_V channels.

6.3.3.4 Debye Layer Extracellular Potential

The most interesting part for the myelinated axon certainly is the extracellular potential. One of the main questions is whether the AP echo observed in the EAP of the unmyelinated axon will be damped out by the introduction of a myelin insulation.

The Debye layer extracellular potential at nodes of Ranvier clearly shows the same shape as the intracellular potential in fig. 6.12, thereby not displaying a different behavior than in the unmyelinated case (cf fig. 5.11 on page 86). They even show the same amplitude, which is not too surprising – the membrane has the same thickness and permittivity at nodes of Ranvier as in the unmyelinated case, and the intracellular potential amplitude has a comparable size as well, so a comparable extracellular potential according to eq. (5.5) on page 85 could be expected.

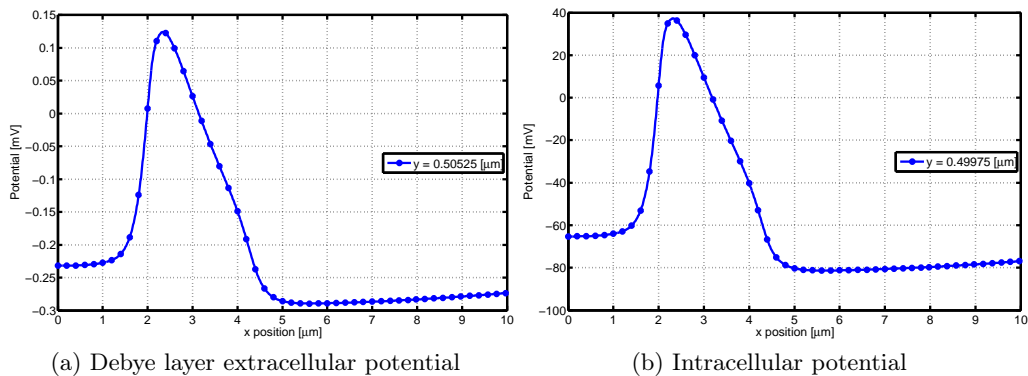


Figure 6.12: **Debye layer potential time course at node of Ranvier.** The Debye layer potential directly at the extracellular membrane interface (*left*) can be seen to directly correspond to the intracellular potential on the opposite membrane interface (*right*), the only difference being the amplitude, which was significantly reduced by the membrane.

This does, of course, not hold for the myelin parts of the membrane, where the electric properties of the membrane have changed significantly. The result can be seen in fig. 6.13, where the effects of the lower myelin permittivity can be seen in the greatly reduced amplitude compared to the node of Ranvier in fig. 6.12a.

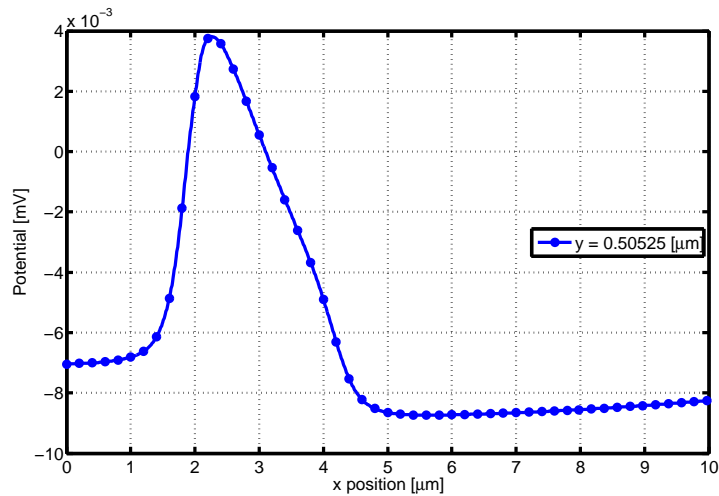


Figure 6.13: **Debye layer potential time course at myelin.** The effects of the lower membrane permittivity are clearly visible, as the potential amplitude is greatly reduced compared to the one in front of a node of Ranvier in fig. 6.12a

6.3.3.5 Near- and Farfield Extracellular Potential

To get a better idea of how the extracellular potential looks like outside the Debye layer, the potential time courses at various positions were plotted in fig. 6.14. The arrangements of plots corresponds to the computational domain, i.e. plots in the same row have the same y -coordinate, and plots within the same column have identical x -coordinates. The second column has its x -value at a node of Ranvier, while the first and third columns are located at the center of the neighboring internode (myelin) segments. The difference in the EAP curves close to the membrane is striking (bottom row), while at more distant locations (middle rows) the time courses at myelin and node positions become more and more alike. At a distance of a few hundred micrometers from the membrane (top row), the signals look the same for each x -coordinate.

The interpretation is straightforward: close to the membrane, the potential mostly “sees” what is happening in the membrane compartments close-by and is therefore dominated by its activity. This explains why nearfield potentials at myelin and nodes show such distinct shapes: at nodes, membrane currents are strongest and dominate the potential. Conversely, at myelin compartments, there are no membrane currents at all, so the potential just shows the damped AP echo

– albeit with a greatly reduced amplitude, which can be read from the scale of the y -axis between e.g. the bottom-left and bottom-middle plot.

The further away from the membrane, the more important become contributions from other membrane parts, resulting in a harmonization of potential curves at different x positions. Interestingly, the sheer distance dependence of the Poisson equation makes for a “democratization” of the membrane contributions. The strong contributions of the small-extent nodes of Ranvier and the small contributions of the much longer myelin parts become equally important in the distant extracellular field.

This also means that even in the myelinated case, the AP echo and the resulting capacitive currents represent an important contribution to the EAP, which was not intuitively clear from the beginning.

6.4 Comparison with LSA

Following up on the comparison of EAPs at different distances from the membrane in section 5.3.1, we now do the same for the myelinated axon. The interesting question now is if we can find the same qualitative results, i.e. a good agreement at larger distances and significant deviations close to the membrane. Since now the axon is not homogeneous, we have to compare the potential time courses not only at different y distances, but also at different x -coordinates. For the total membrane current in eq. (5.6) on page 93, this also involves replacing the previously constant membrane capacitance by a position-dependent capacitance $C(x)$, because now $A(x)$, $l(x)$ and $\epsilon_{\text{memb}} = \epsilon_{\text{memb}}(x)$ vary with x in eqs. (5.2) and (5.3). For the only free parameter in the LSA eq. (2.40), the same value as before was chosen for the resistivity $\rho = 72 \Omega \text{ cm}$.

In fig. 6.15, the time courses of the extracellular potentials are shown in a grid representation, where the position of each subplot roughly gives its position in the actual computational grid, as before in fig. 6.14. Again, points close to the membrane are found in the lower rows, points with larger distances in the upper rows. The arrangement of columns and the positions of the individual subplots are the same as in fig. 6.14.

For larger distances as in fig. 6.15, the overall agreement between electrodiffusion and LSA models is generally good, albeit in contrast to the unmyelinated case, notable differences can also be found in the farfield, e.g. at peak P1 or between peaks N1 and P3. Another comparison for the nearfield can be found in fig. 6.16, which shows a zoom into the transition interval ($x \in X^{(\text{tran})}$, middle columns) between myelin parts ($x \in X^{(\text{myel})}$, left columns) and one node of Ranvier

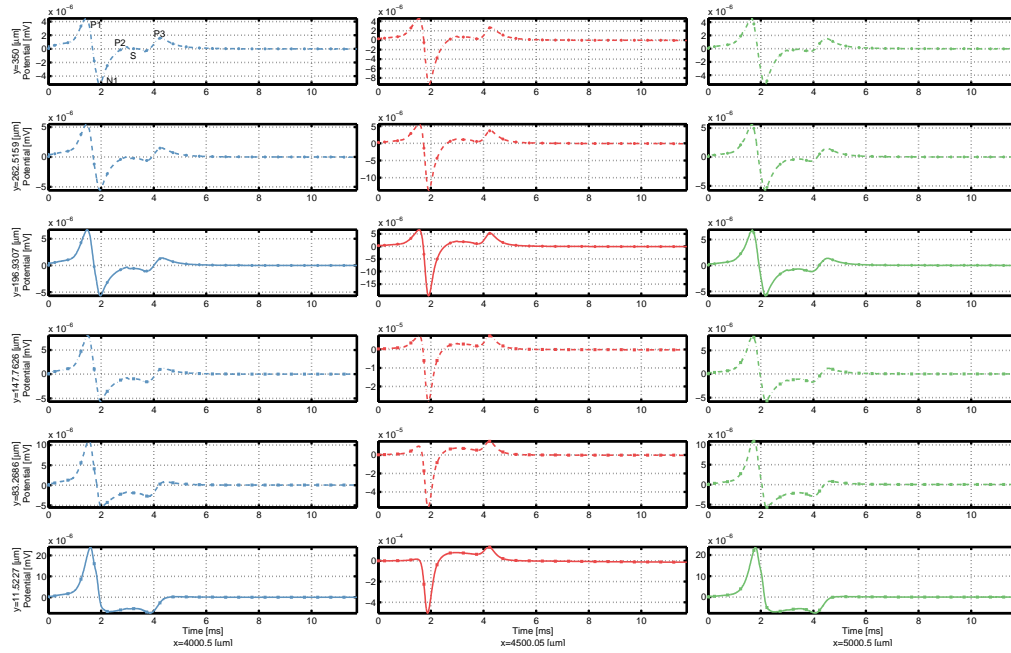


Figure 6.14: **Extracellular potential time courses at various positions.** Plots in the same row share a common y -coordinate and in the same column a common x -coordinate. They are arranged by ascending x -coordinates to the right and by ascending y -coordinates in the upward direction. Columns 1 and 3 correspond to myelin segments, column 2 to a node of Ranvier. While potential shapes are distinct close to the membrane in the lower rows, they become more and more similar in the upper rows, showing the distance dependence of contributions from different membrane compartments to the extracellular field. The different features of the characteristic triphasic shape are labeled as in the previous chapter, shown only for the upper left subplot.

6 Model of a Myelinated Axon

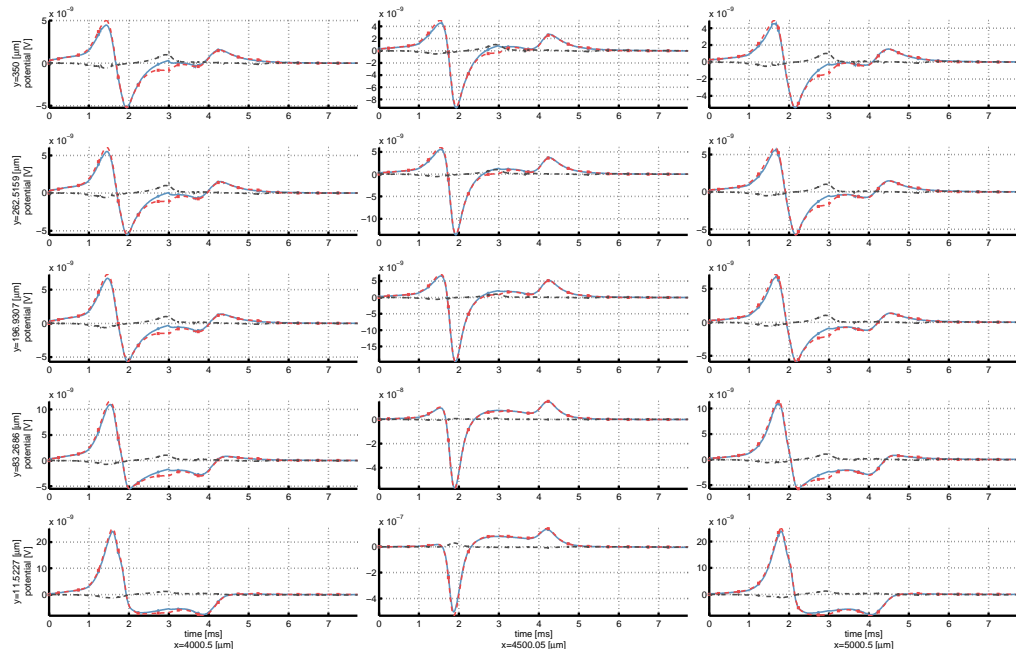


Figure 6.15: **Comparison with LSA at various farfield positions.** The same potential time courses as in fig. 6.14 (*solid lines*) are compared with the LSA potential (*dashed lines*). Additionally, the difference between both curves is shown (*dash-dotted lines*). The agreement between both models generally is very good and shows notable deviations only between peaks N1 and P3.

($x \in X^{(\text{node})}$, rightmost column). Here, some interesting details can be made out. Firstly, large parts of the plots at myelin coordinates show very good agreement, even for distances smaller than $5\ \mu\text{m}$. At distances $< 1\ \mu\text{m}$ and especially in the Debye layer (bottom row), however, the differences are large. On the other hand, the node of Ranvier column (right) shows stronger deviations, even up to distances of $5\ \mu\text{m}$. This is consistent with our previous observation from section 5.3.1.1, where the ionic contributions accounted for the greatest deviations between electrodiffusion and LSA models. Following this reasoning, it is clear that the nodes of Ranvier with their large ionic currents and comparably small capacitive currents will show stronger deviations than myelin parts, which only contribute a single capacitive EAP component.

Figures 6.15 and 6.16 actually give insights into the “big picture” of the EAP generation. It shows the distinct shapes due to the domination of either ionic membrane currents (nodes of Ranvier) or capacitive effects (myelin). Furthermore, it shows how those two shapes merge at greater distances from the membrane to constitute the characteristic triphasic “up-down-up” shape of the farfield LFP.

6.5 Summary

The introduction of myelin required some fundamental changes in the model. By recognizing that the transmembrane potential is essentially a linear function, we could avoid explicitly representing compartments with varying membrane thickness in the mesh. Instead, myelin could be represented implicitly through adapted effective dielectric constants. With this, we maintained the tensor grid structure in the spatial discretization, which is very beneficial for the computational efficiency.

The problem of jumping material coefficients between myelin and nodes of Ranvier was tackled in two stages: by resolving the transitions in x -direction geometrically in the grid, and by applying an additional smoothing operator to the discontinuity in permittivities. We take the view that smoothing the permittivities is actually more realistic, since in real neurons (cf. again the illustration in fig. 2.1 on page 25), the internodes do not show an abrupt increase in membrane thickness, but rather a gradual one, accompanied by a corresponding smoother change in the dielectric constant.

The presented changes in model and numerical methods allowed us to simulate a myelinated axon of length 4.8 cm with 48 nodes of Ranvier. The results show the expected AP velocity increase and two different forms of extracellular potentials: one that is dominated by membrane currents at nodes of Ranvier,

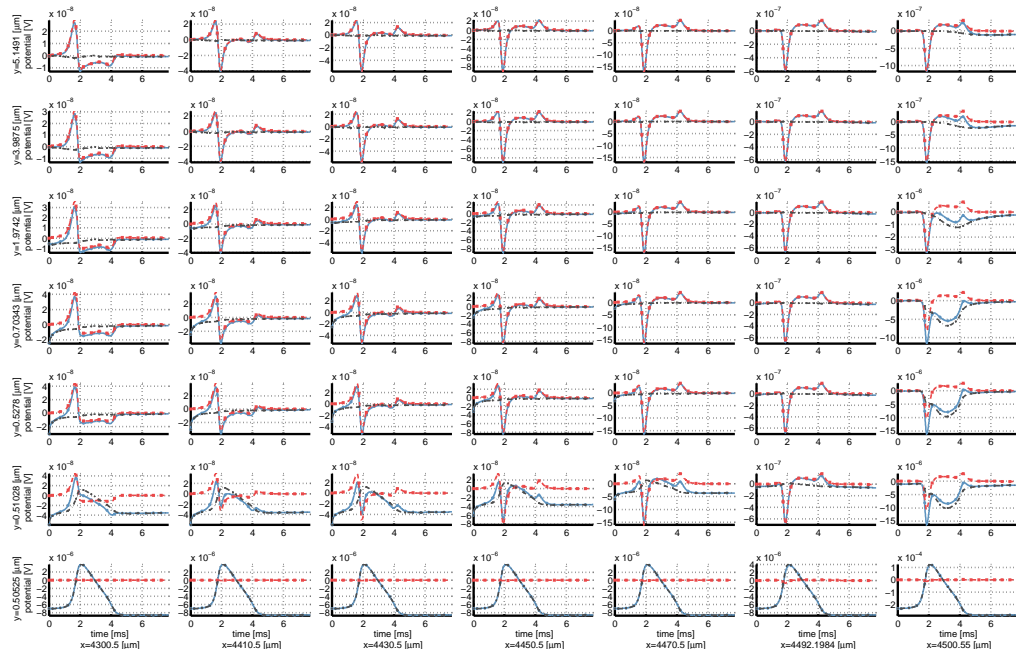


Figure 6.16: **Comparison with LSA: zoom into myelin-node transition and nearfield.** A closer look at the transition between myelin (*left*) and node of Ranvier (*right*) membrane parts in y -direction, plotted here for nearfield distances up to about $5 \mu\text{m}$, reveals significant deviations (*dash-dotted line*) between ED (*solid lines*) and LSA (*dashed lines*) especially at nodes of Ranvier. Columns in between mark coordinates in the transition interval between myelin and node of Ranvier. Myelin coordinates show strong deviations only up to a distance of about $1 \mu\text{m}$.

and one that is purely capacitive at internodes. Interestingly, these two different **EAP** shapes merge at larger distances and show the same triphasic shape as an unmyelinated axon. This can be explained by the interplay of **EAP** amplitude, which is considerably higher at nodes, and of the contributing membrane area, which is significantly higher for internodes. At sufficiently large distances, the weighting of these two components balances and both components – nodal ionic and internodal capacitive – become equally important.

A comparison with the **LSA** model as before showed deviations especially at small membrane distances, which could again to a large degree be attributed to the ionic components of the **EAP** and their influence within the intermediate diffusion layer. The extracellular potential dominated by capacitive currents at internodes generally shows a very good agreement. As before, an exception is the Debye layer, which exhibits fundamental differences in comparisons with **LSA**.

7 Ephaptic Interaction Between Multiple Axon Fibers

If you're going through hell, keep going.

(Winston Churchill)

Now that the basic models for unmyelinated and myelinated nerve fibers have been derived, simulated, and validated, we would like to study interactions between nerve fibers. Of particular interest is the phenomenon of *ephaptic coupling*, which describes the influence of one neuron on the other in the absence of chemical or electrical synapses (*gap junctions*). Such a coupling is possible either directly through the electric field or indirectly by exchange of ions. In either case, the neurons have to be very close to each other to yield a notable effect, which is reflected in the adjective *ephaptic* (from Greek $\epsilon\phi\alpha\pi\tau\omega$, “to touch”) and the associated site *ephapse*, the “location of close contact or vicinity” [6], in distinction to a synapse.

7.1 Previous Work

While both chemical and electric synapses are traditionally believed to be the only possible source for eliciting an action potential, there is a surprisingly large body of studies on the ephaptic effects between two neighboring fibers, starting in the 1940s [62] and continuing until today, also involving the LFP of a population of neurons as a possible source for AP synchronization [5]. An excellent overview on the existing literature with slightly different emphases is given in [40] and [59]. A newer review is [103], which focuses more on the electric fields of neuronal populations and their functional effects.

A special case in this context is the heart muscle, in particular *cardiomyocytes* (cardiac muscle cells), which are considered to be excited via gap junctions [3, 48]. In [91], a modeling study was able to show that gap junctions are not necessary, since ephaptic coupling may serve to conduct the AP from cell to cell over the cleft between cells. In an experimental study using mice with suppressed expression of myocardial gap junctions [47], the animals were found to develop

cardiac arrhythmia and eventually died by sudden cardiac arrest at about two months of age. Until then, however, they showed normal heart development and function, indicating that the basic conduction functionality was maintained in the absence of gap junctions.

A number of modeling studies have addressed the topic of ephaptic coupling. Holt [56] found the influence of the extracellular electric field of one axon onto the other to be a fraction of a millivolt, which he considered too small to elicit an AP in a cell which is not already very close to threshold. He used the LSA model for this, neglecting concentration effects. Mori [73], on the other hand, used a reduced version of the electrodiffusion equations (the “electroneutral model”) and studied ephaptic conduction in the setting of myocardial fibers mentioned above. He was able to demonstrate that ephaptic coupling indeed can supplement or replace gap junction conduction [75]. However, the cleft width had to be at most 5 nm in order for the conduction to be successful.

Holt [56, chapter 2.3.1] also mentions a number of studies that address the interaction between neighboring axons with respect to phase-locking. Essentially, APs running in two parallel axons will tend to synchronize: if they are aligned, the extracellular current sources and sinks are also aligned, and each axon will try to hyperpolarize the other, resulting in a slowed-down AP conduction. In staggered APs on the other hand, the outward currents of the leading axon will align with the inward currents of the following axon, which will enhance the following axon’s AP conduction. As a consequence, staggered, phase-locked APs are stable and faster, while aligned AP will slow each other down.

In the following chapter, we will use the electrodiffusion model developed over the course of the previous chapters to study ephaptic coupling between parallel axons in a relatively simple setting.

7.2 Unmyelinated Axon Surrounded by an Axon Bundle

7.2.1 Geometry

Due to the cylinder symmetry, we are limited in the choice of our geometry, since the angular direction was eliminated from the equations. This means we can not model two single axons next to each other, which is best exemplified in a picture, see fig. 7.1.

It is clear that, when adding a second axon, this can not be modeled with one single membrane anymore, as we can only use the symmetry axis once. Two

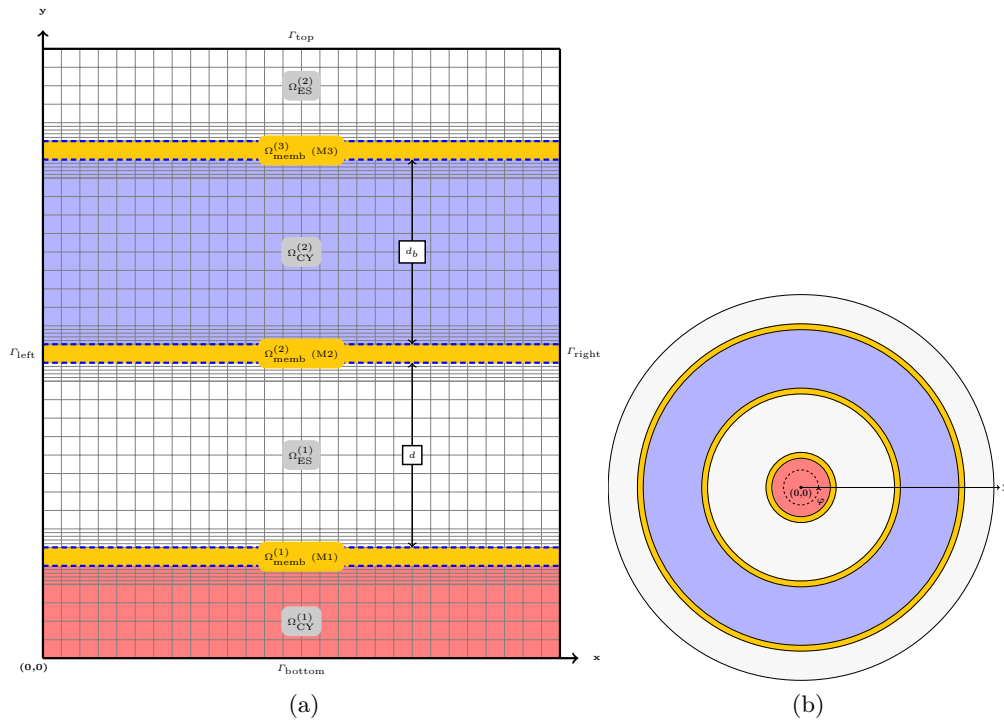


Figure 7.1: **Computational domain for multiple axons.** The two-dimensional computational domain (*left*) was modified in order to include two additional membranes M2 and M3, enclosing a second cytosol domain. On the right, the equivalent cylinder cross section for this setup is shown, revealing a central axon that is surrounded by an interstitial space and an axon bundle, which is again surrounded by an extracellular space.

additional membranes are needed to obtain an additional cytosol subdomain $\Omega_{\text{CY}}^{(2)}$ next to the obligatory $\Omega_{\text{CY}}^{(1)}$, as well as an additional subdomain $\Omega_{\text{ES}}^{(2)}$, as shown in fig. 7.1a. $\Omega_{\text{ES}}^{(1)}$ is now bounded by two membranes M1 and M2. In the cylinder geometry, however, this does not correspond to two adjacent axons, but rather to one cylindrical axon surrounded by a shallow cylinder axon (see the cross section in fig. 7.1b).

We can think of this as an *axon bundle* around a single axon. It is clear that this situation is not realistic, since in real tissue the axon bundle would not consist of a single intracellular domain, but of multiple cylindrical axons with a certain spacing in between them. We should keep this in mind when interpreting the results generated by this model and consider them as (at least partially) speculative. For a basic phenomenological study, however, this setup shall prove useful.

A similar geometry is used in *core conductor models* [85]. When an axon is surrounded by a resistive sheath, as is often the case in peripheral nerves, e.g. by an *epineurium* or *perineurium*, the extracellular space can be regarded to be one-dimensional, in analogy to the cable equation [29]. This approximation, however, only holds for small distances between axon and sheath. Moreover, here we are not only interested in the extracellular field due to a restricted extracellular volume, but also on its impact on adjacent nerve fibers.

A crucial parameter is the distance between axon and bundle, d . Holt [56] considered the case of an axon bundle and estimated the spacing d based on different arrangements of axons, shown as cross-sections in fig. 7.2. In a rectangular assembly, as in fig. 7.2a, the packing is non-optimal. The volume fraction

$$\alpha = \frac{V_{\text{ES}}}{V_{\text{total}}} \quad (7.1)$$

between extracellular volume V_{ES} and the total volume V_{total} can be calculated as

$$\alpha_r = \frac{(2r + 2d)^2 - \pi r^2}{\pi r^2} = \frac{4r^2 + 8rd + 4d^2 - \pi r^2}{\pi r^2}.$$

Even for touching axons with $d = 0$, this gives a value of $\alpha_r = \frac{4-\pi}{\pi} \approx 0.2146$, which is larger than the observed average volume fraction of $\alpha = 0.2$ [94]. When using a hexagonal packing as in fig. 7.2b, which Gauß proved to be optimal (see

[92]), the volume fraction reads

$$\alpha_h = \frac{a^2 \frac{3}{2} \sqrt{3} - 3\pi r^2}{a^2 \frac{3}{2} \sqrt{3}} = \frac{(2r + d)^2 \frac{3}{2} \sqrt{3} - 3\pi r^2}{(2r + d)^2 \frac{3}{2} \sqrt{3}}$$

with the hexagon side length $a = 2r + d$. For $d = 0$, this results in the densest possible packing with an extracellular volume ratio of $\alpha_h = \frac{2\sqrt{3}-\pi}{2\sqrt{3}} \approx 0.0931$. In the present case of an axon radius of $r = 500$ nm, a volume ratio of 0.2 would be obtained for a spacing of $d \approx 65$ nm, which is already on the upper bound of the estimated average extracellular width of 38 nm–64 nm [94]. This average membrane-to-membrane distance is remarkably small, and – considering the results of the preceding chapters – concentration effects may be assumed to play an important role within this regime, which is only one order of magnitude larger than the Debye length γ_D , and definitely within the diffusion layer regime of order $\sqrt{\gamma_D}$ (cf. [73]).

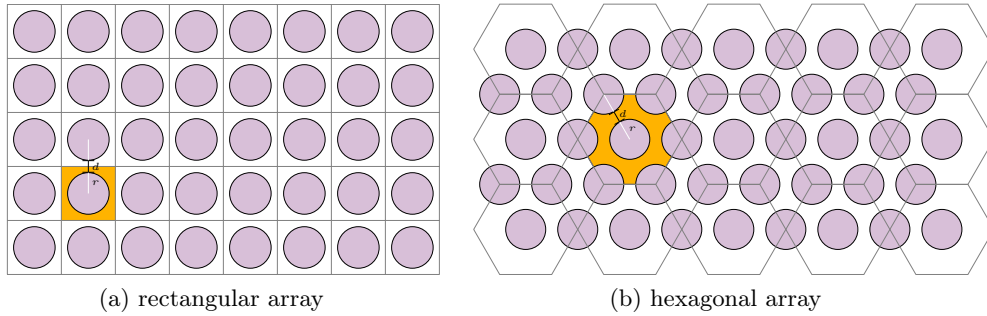


Figure 7.2: **Illustration of axonal packing in different geometric arrangements.** A regular rectangular packing (*left*) leads to a non-optimal packing ratio. The optimal packing can be achieved in a hexagonal array (*right*).

7.2.2 Action Potential

In an attempt to quantify the influence of one axon onto the surrounding bundle, an action potential was elicited in the central (unmyelinated) axon as before, while the axon bundle was not stimulated. We largely used the same parameter set as in chapter 5, as listed in table 5.1 on page 70, with the exception of an adjusted mesh according to the changed geometry. As in the previous setups,

the Debye layers of all three membranes were fully resolved, and coarsening strategies were employed for the regions in between to minimize the total number of unknowns.

Simulations were carried out for various inter-fiber distances $d = 1 \mu\text{m}$, 500 nm, 100 nm and 50 nm. Distances were deliberately chosen larger in tendency than the average ES width of 38 nm–64 nm in order to obtain an estimated upper bound on d for which ephaptic interactions are possible.

The diameter of the axon bundle d_b was chosen in two different ways: in the first set of simulations, it was fixed at $d_b = 500 \text{ nm}$, which means that for larger inter-fiber distances d , the volume of the axon bundle will be larger than for small distances due to the cylinder geometry. Therefore, we also implemented a second version in which the bundle volume was kept constant (and equal to the source axon volume) by varying d_b dependent on d , which is called the “volume-corrected” setup. We will later on see the reason why the cell volumes and their relationship to the extracellular ones are of significance. Figure 7.3 shows the fraction of the three volumes of interest, i.e. those of the interstitial region $\Omega_{\text{ES}}^{(1)}$, and of the two intracellular domains $\Omega_{\text{CY}}^{(1)}$ and $\Omega_{\text{CY}}^{(2)}$, for the two different variants.

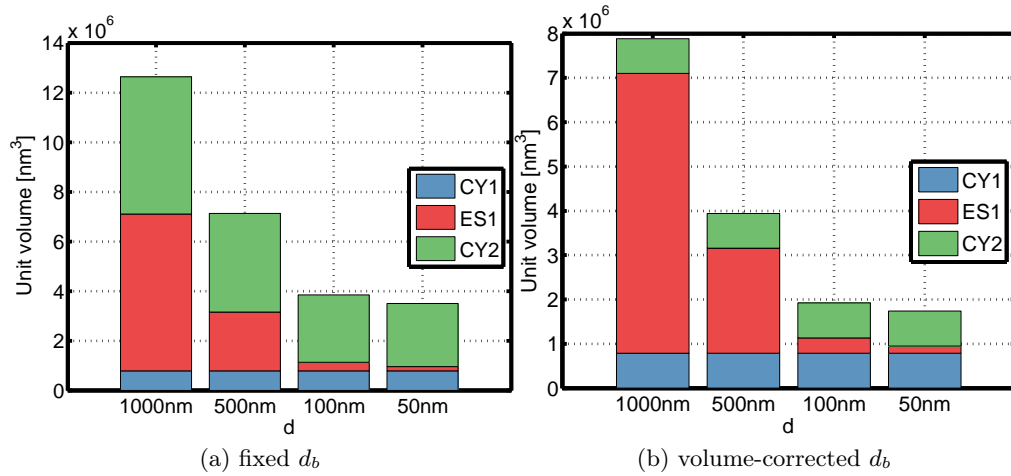


Figure 7.3: **Ratios of intra- and extracellular volumes for different axon-bundle distances.** In the fixed case (*left*), the bundle volumes are growing with inter-fiber distance, thereby also changing relative volume fractions. In the volume-corrected case (*right*), the bundle volume is kept fixed and equal to the central axon volume, such that only the interstitial volume in between is changing with d .

7.2.2.1 Fixed bundle diameter d_b

Table 7.1 shows the computation times¹ for each of the different choices of d , obtained from a parallel run with $p = 10$ processors. The overall performance is reasonable – the smaller the distance, the more interaction between fibers, increasing the difficulty of the numerical problem.

Table 7.1: **Simulation timings for the axon bundle setup using different inter-fiber distances d and a fixed bundle diameter d_b .** For each distance d , the number of unknowns, total computation time, the needed number of time steps, and the average solution time per time step (full Newton iteration) are shown.

d [nm]	# DOFs	Total comp. time [s]	# time steps	avg. time / time step [s]
1000	116352	156071	2026	77.034
500	114332	164591	2026	81.24
100	114332	181175	2026	89.425
50	114332	184747	2026	91.188

Figure 7.4 shows the membrane potential at each of the three membranes, for different distances d . At a larger distance of 1 μm , the effect on the axon bundle is small. Already at a distance of 500 nm, however, the axon bundle spikes in response to the source axon’s AP. Interestingly, the outer membrane M3 fires with a notable delay towards the inner membrane M2.

In a real geometry, the outer axons would, of course, not have two distinct membranes, but outer and inner membrane would be a single connected structure. By this direct coupling, the axonal membrane would probably fire as a whole instead of having a time delay between inner and outer parts of the membrane, consistent with the “all-or-none” principle of action potential generation.

As axon and bundle come even closer in fig. 7.4c and fig. 7.4d, the picture changes. Membrane M2 fires earlier, but membrane M3 does not fire anymore. Instead, membranes M1 and M2 generate an additional AP of reduced amplitude, which is even smaller for the case $d = 50$ nm.

These results are quite remarkable. How can they be explained? For this, we

¹Note that the absolute computation times are rather high in comparison to chapter 5. Accidentally, these simulations have been run using an executable that was compiled completely without optimizations, significantly impairing the performance. In table A.1 on page 159, the effect of using compiler optimizations can be clearly seen.

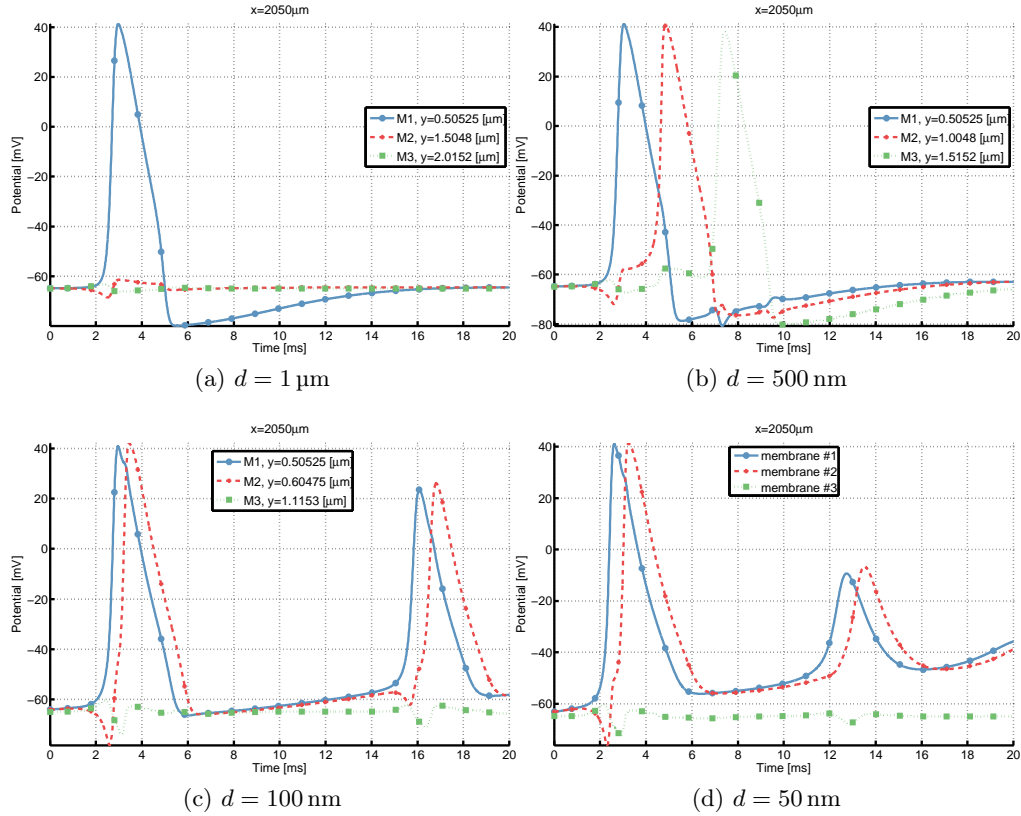


Figure 7.4: **Membrane potentials for different axon distances d and fixed bundle diameter d_b .** For decreasing distances d , the ephaptic interactions of axon and bundle changes. For $d = 1000 \text{ nm}$, only sub-threshold potentials are induced. For smaller distances, an AP is elicited in the two bundle membranes. For $d \leq 100 \text{ nm}$, this even leads to an oscillation between axon and bundle. For $d = 50 \text{ nm}$, the membrane potential drifts away from resting potential, as the extracellular concentration balance is disturbed.

look at the intra- and extracellular potential and charge density time courses directly adjacent to the three membranes.

For $d = 1 \mu\text{m}$, the intracellular potential in fig. 7.5c takes on the expected form of an action potential, albeit with a reduced amplitude in comparison to the isolated unmyelinated axon from fig. 5.4 on page 78. A different picture shows for the extracellular potential in fig. 7.5d, which is not determined by the AP echo, as in the case of an unrestricted extracellular space (cf. fig. 5.11 on page 86). Instead, it reaches an amplitude of several millivolt. It is clear that the relatively large extracellular potential contributes significantly to the total membrane potential in fig. 7.5a. Previously, the membrane potential was almost exclusively determined by the intracellular potential. Now, the intracellular potential still contributes most of the membrane potential amplitude, but notable -5 mV are provided by the extracellular potential.

We note that it is the membrane potential *difference* $[\phi] = \phi_{\text{in}} - \phi_{\text{out}}$ which is “seen” by the ion channels, such that a depolarization of the cell can be reached by either an increase in the intracellular potential ϕ_{in} or by a decrease in the extracellular potential ϕ_{out} .

How much of the membrane potential is contributed by the intracellular or extracellular regime here depends to a large degree on the ratio of intra- and extracellular volumes, see fig. 7.3. We see that the volume of $\Omega_{\text{ES}}^{(1)}$ is larger, but within the same order of magnitude as those of the two intracellular domains $\Omega_{\text{CY}}^{(1)}$ and $\Omega_{\text{CY}}^{(2)}$. In smaller volumes, the same amount of charge results in a larger average charge density, i.e. the right-hand side of the Poisson eq. (2.43), and will therefore cause a larger potential.

We once again acknowledge that the electric potential inherently is a *relative* measure, such that a certain membrane potential difference, as generated by the HH system, will be distributed differently onto the intra- and extracellular spaces depending on the respective volumes. In chapter 5, the intracellular volume was negligible in comparison with the extracellular one, so the membrane potential was approximately equal to the intracellular potential. Now, the ratio between intra- and extracellular volumes is not negligible, and consequently part of the membrane potential is shifted to the extracellular potential.

We emphasize that this reasoning only applies to our very special geometry with a severely confined extracellular space, greatly reducing the extracellular conductivity. In realistic geometries, the ES is generally not isolated, but highly connected, such that the extracellular conductivity is not reduced proportionally with volume.

The influence of this large extracellular potential on the adjacent bundle can be

seen in fig. 7.5a already at this relatively large distance. The bundle's membrane potential at membrane M2 responds with an initial hyperpolarization, followed by a depolarization of about the same size. This pattern of a hyperpolarization followed by a depolarization in ephaptic interactions has been reported before using a different model [9]. It is the consequence of the extracellular potential with its initial rise and the following trough, as visible in fig. 7.5d.

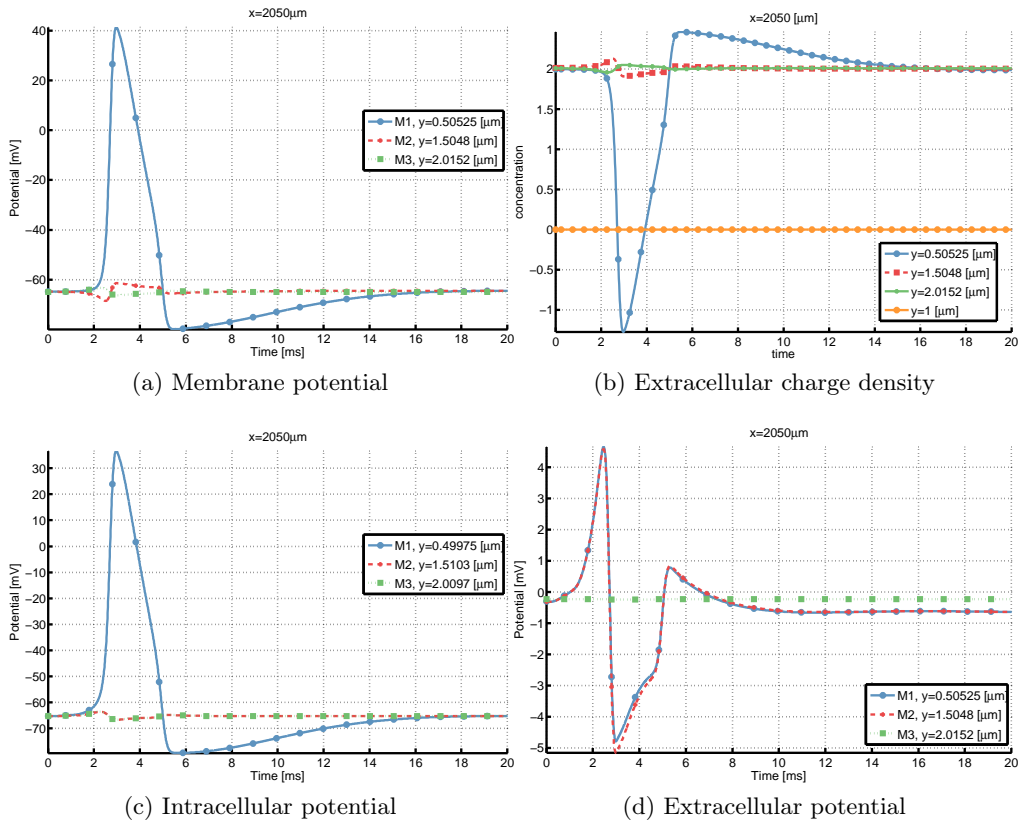


Figure 7.5: **Potential and charge density time courses for $d = 1 \mu\text{m}$.**

This becomes even more obvious if d gets smaller, as the ratio between intra- and extracellular volumes further increases, as can be seen in fig. 7.6. The amplitude of the intracellular potential of the central axon is further reduced, and the EAP amplitude next to membrane M1 is increased to about -10 mV for now. The extracellular potential at membrane M2 is the same, since it is not significantly reduced over the small range of $d = 500 \text{ nm}$. This large EAP first

causes a hyperpolarization of membrane M2, and then a depolarization, which is strong enough to elicit an action potential, as visible in fig. 7.6a. At membrane M2, the ratio of intra- to extracellular potentials now is completely reversed, in accordance with the volume ratio: the extracellular volume is tiny compared to the intracellular volume of the axon bundle.

As a result, the depolarization of membrane M2 is mainly not achieved by an increase of the intracellular potential, but through a huge “inverse spike” in the extracellular potential with an amplitude of about -100 mV. However, the small intracellular response is enough to trigger an action potential also on the opposite membrane M3, which fires in the normal mode, by an increase in the intracellular potential. The giant EAP generated by the second AP is also visible in the intracellular potential of the central axon as a large hyperpolarization.

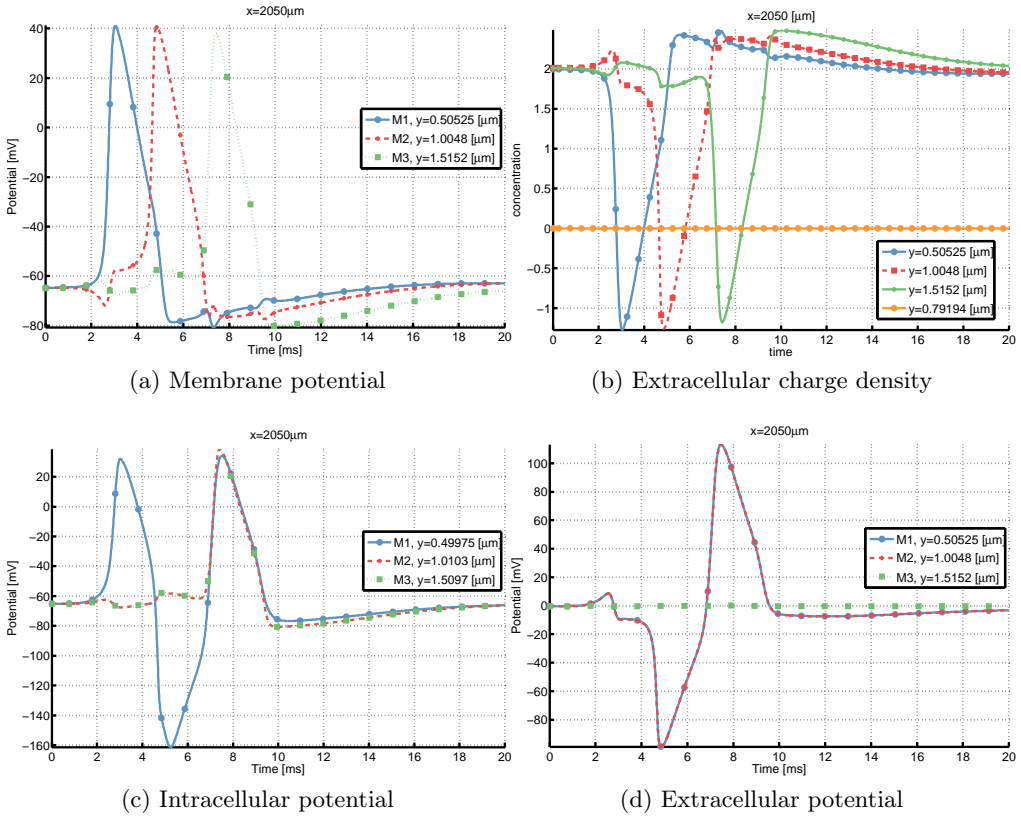


Figure 7.6: **Potential and charge density time courses for $d = 500$ nm.**

At an even smaller distance of $d = 100$ nm, the extracellular volume is now significantly smaller than the central axon's volume, such that the AP is completely generated by a large inverse spike in the extracellular space in fig. 7.7, as before at membrane M2. Consequently, an AP is elicited also at membrane M2 through the large EAP, after an initial hyperpolarization.

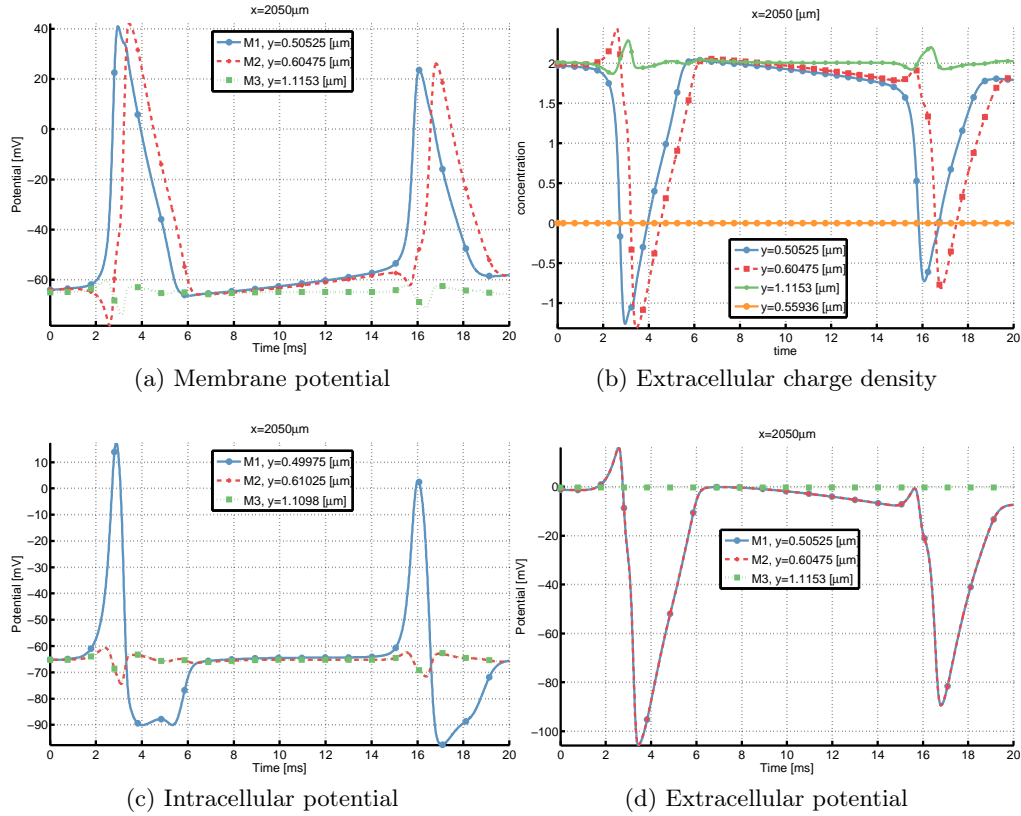


Figure 7.7: **Potential and charge density time courses for $d = 100$ nm.**

Since the volume ratio between $\Omega_{ES}^{(1)}$ and $\Omega_{CY}^{(2)}$ is now highly in favor of the axon bundle, the intracellular response to the AP is even smaller. This response is not sufficient to depolarize membrane M3 enough to trigger an AP there. But another interesting effect can be observed: the effect of the “hyperpolarization phase” in the inverse extracellular spike is large enough to depolarize both membranes M1 and M2 and let them fire once again, albeit with a smaller amplitude.

For the case $d = 50$ nm, basically the same qualitative situation as for $d =$

100 nm can be observed. However, the extracellular volume is now so small that an undersupply of concentrations is formed. As a result, the membrane potentials at M1 and M2 do not fully return to their resting potential in fig. 7.8a. Two spikelets are generated afterwards, and the membrane potential then further drifts away from its resting potential. The concentration balance is now disrupted completely.

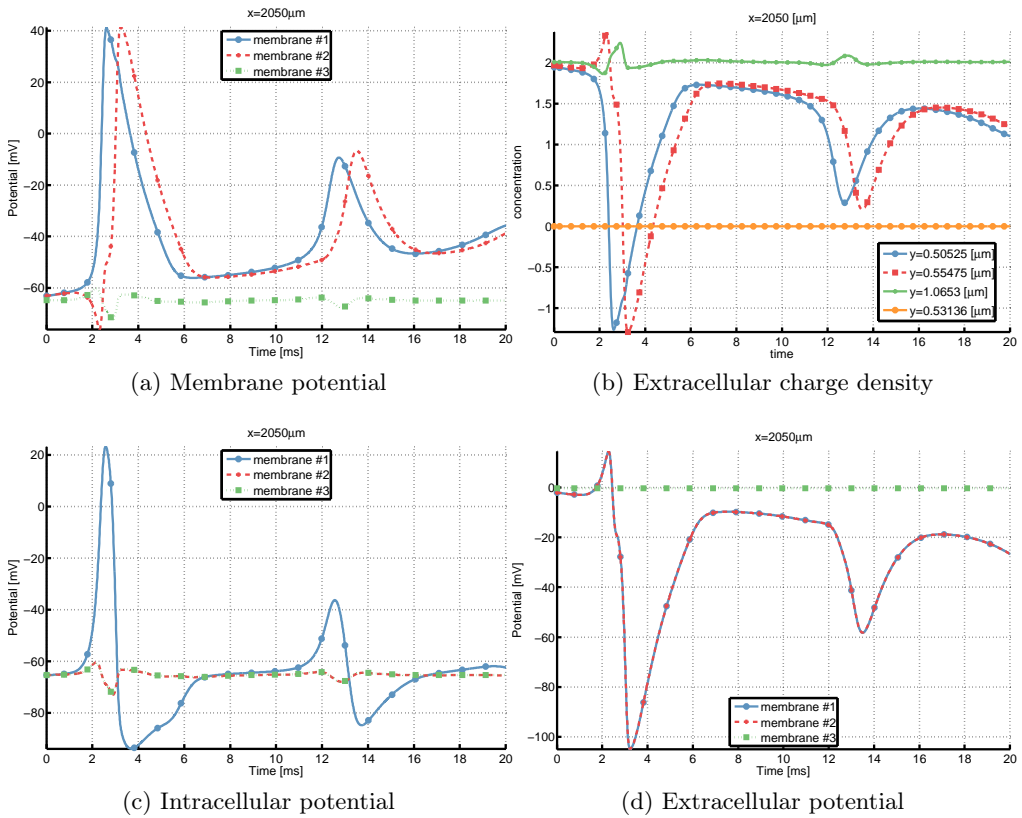


Figure 7.8: Potential and charge density time courses for $d = 50$ nm.

7.2.2.2 Volume-corrected bundle diameter d_b

The results for the volume-corrected axon bundle in fig. 7.9 basically show the same picture as the previously shown findings for varying axon bundle volumes. Some minor differences can be made out, e.g. the reduced delay between membrane M2 and M3 for the case $d = 500$ nm due to the smaller distance d_b .

7.2 Unmyelinated Axon Surrounded by an Axon Bundle

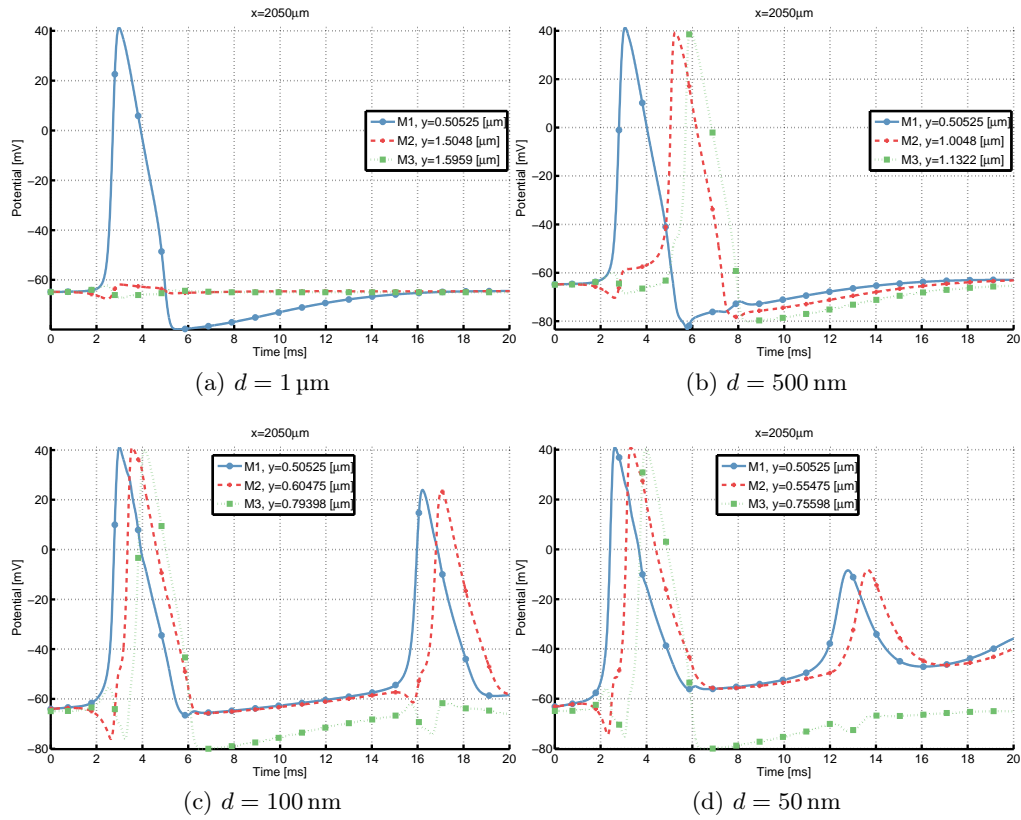


Figure 7.9: Membrane potentials for different axon distances d and adjusted bundle diameters d_b (volume-corrected).

According to the elaboration above, the ratio of volumes influences the distribution of membrane potential onto the intracellular and the extracellular potential contribution. Indeed, the intra- and extracellular potential time courses look completely different than those for the fixed d_b , although the membrane potential shows a similar behavior. We refrain from showing the detailed results here and defer this to appendix A on page 159, see figs. A.1 to A.3.

7.3 Summary

In this chapter, we used a somewhat artificial setting to study ephaptic effects of one axon on a surrounding axon bundle through an extremely isolated extracellular space. Quite remarkably, the emerging extracellular potentials are large – even larger than the intracellular ones – due to the small extracellular volume fraction and the resulting high resistivity. An action potential traveling along the central axon was able to elicit another AP in the surrounding bundle already at a distance of 500 nm, a multiple of the actual average cell-to-cell distance in the brain. In a more realistic representation, the extracellular space would, of course, be connected and the resistivity would therefore be lower. Accordingly, the maximum distance for inducing an AP in another cell by ephaptic coupling would probably be much lower.

Nevertheless, some interesting effects can be seen in the above results. In some configurations, an oscillation between the two fibers can be seen, and the bulk concentrations in the extracellular volumes are thrown out of balance for small distances. The following discussion will try to put these results into the experimental context and estimate its physiological relevance.

8 Summary and Discussion

Ich habe mich vertan.

(Helge Schneider)

In this work we have set up the mathematical model of axon fibers embedded in extracellular space based on the [PNP](#) equations of electrodiffusion. A suitable method for the numerical simulation of this nonlinearly coupled system of [PDEs](#) has been presented. It is designed to properly represent the crucial physical properties of the underlying equations in the numerical scheme. The key steps to yield a stable and efficient algorithm were:

- The grid had to resolve the Debye layer close to the membrane in the normal direction. For the x -direction, a much coarser mesh size was sufficient.
- The [PNP](#) system had to be solved in a fully-coupled fashion, as an operator-splitting approach dramatically reduced the time step size needed for computing a non-oscillatory solution.
- The choice of an implicit time-stepping scheme took advantage of higher stable time step values and accounted for the diffusion-dominance in the Nernst-Planck equation, which we observed for parameters in the physiological range.
- The system had to be carefully equilibrated before setting a stimulus, as the concentration profile towards the membrane had to reach steady-state in order to get meaningful reversal potentials for each ion channel.
- The application of a *threshold volume scaling* was crucial to compensate the large differences of residual magnitudes introduced by the strongly varying cell volumes in this cylinder geometry. With this, the Newton iteration was able to converge even for large domain sizes with an only slightly lower average time step size compared to the 2D problem.
- The blocking of unknowns into dense 4x4 matrices at each grid vertex turned out to be mandatory for large numbers of [DOFs](#), since it enabled the exact inversion of diagonal blocks by the [ILU](#) preconditioner.

- A suitable choice of linear solvers and matching preconditioners was critical for the convergence and efficiency of the algorithm, especially in the parallel case. The application of state-of-the-art iterative solvers was a necessary requirement to cope with the grid anisotropy and to enable parallelization on larger processor counts.

We emphasize that, when taking into account the above measures, the [PNP](#) could be solved numerically with standard methods, i.e. with conforming linear finite elements for the spatial and an implicit Euler method for the time discretization, despite the multi-scale character of the underlying model.

For the representation of myelin, effective permittivities were introduced. The permittivities were scaled to correct for the increased myelin thickness of myelin sheaths. The benefit is that the same tensorial grid structure as for the unmyelinated axon can be used. The grid in y -direction did not have to be changed at all. The x -direction, of course, had to be adapted to represent nodes of Ranvier in correct physical dimensions. Furthermore, permittivities were smoothed at transitions between myelin and nodes of Ranvier. This not only suppressed numerical oscillations at these transition zones due to a permittivity jump, it also represents the actual geometry of myelin sheaths, which smoothly increase in width near nodes of Ranvier instead of showing an abrupt change of thickness.

By means of this numerical solution strategy, we were able to simulate an action potential traveling along an active axonal membrane and its effects on the electric field and ion redistribution in the extracellular space. In an effort to represent all relevant scales of the [PNP](#) model, the Debye layer was explicitly resolved by the computational grid.

To our knowledge, this is the first application of a 3D electrodiffusion model in the context of neuronal excitation with an explicitly resolved Debye layer, which could be shown to be of capital importance for accurately modeling the [LFP](#).

We do not claim that our model is complete or fully realistic from the biological point of view. Instead, it should be regarded as a first attempt to model the dynamics of neural systems on this detailed scale. Model refinement concerning the detailed structure and properties of the active membrane (channel types and densities, membrane surface charges) will have to be done when comparing the results with experimental data, as here we largely used classical data from the squid giant axon with only two types of voltage-gated ion channels.

Moreover, a complete model of a neuron with a dendritic tree seems highly desirable, as this would also allow to quantify the influence of synaptic currents, which are thought to be the main contributors of [EEG](#) and [LFP](#) signals [26, 79]. In future models, also the representation of a realistic extracellular geometry – either

explicitly or as a homogenized model – should be considered. Particularly, the idealized setup of a single axon fiber in an extracellular space used in chapters 5 and 6 consisting exclusively of a homogeneous fluid will rarely occur in reality. In our approach, the main limitation is the cylinder symmetry, which imposes a serious constraint on the geometry. The setup for neighboring axons in chapter 7 is somewhat artificial, but it is nevertheless useful to estimate the impact of ephaptic coupling between nerve fibers in a setting of a severely confined extracellular space. For complex extracellular geometries, a full 3D model would be needed. This would also allow to use reconstructed geometries by e.g. electron microscopy imaging techniques. However, the computational demand of such setups would be much higher than in the present case, and will probably require the use of massively-parallel simulations on supercomputers. This will require some considerations about tailor-made linear solvers and preconditioners to obtain an optimal scalability of the problem.

Alternatively, using the approximation in [73, 75] on a 3D grid promises to be better tractable. The authors used a special boundary to include Debye layer effects without having to explicitly resolve it in the discretization, also eliminating the concentration variables from the equations. Nevertheless, a full 3D electrodiffusion simulation seems desirable, if only to validate the accuracy of this approximation under physiological conditions.

However, one of Mori’s main results, the postulation of an intermediate diffusion layer between Debye layer and bulk solution on the order of $\sqrt{\gamma_D}$, could already be confirmed by our simulations. We found prominent deviations from volume conductor models for distances of at least 5 μm from the membrane. These are a consequence of violations of the electroneutrality assumption in volume conductor models in the Debye layer, which shaped the extracellular electric field to a large degree also in the adjacent diffusion layer. This has been demonstrated for the unmyelinated case in chapter 5 as well as for the myelinated one in chapter 6. Both cases, unmyelinated and myelinated axon, share a common property: the Debye layer shows an extracellular potential that resembles the membrane potential, only reduced in amplitude, where the reduction is depending on different membrane and electrolyte material parameters. We called this the **AP** echo.

Interestingly, the existence of this echo is also observed in experiments. Several experimentalists from Andreas Draguhn’s lab in Heidelberg have confirmed that they see a sudden switch in potential shape when pushing the recording electrode against the neuronal membrane like it is done in juxtacellular recordings (Martin Both, Florian Böhner, Christian Thome, personal communication). This phenomenon can now be explained by the Debye layer dynamics present in our model.

At large distances, results were in good agreement with volume conductor models, consistent with an experimental validation [45]. The fact that the LSA model is not a good representation at closer membrane distances has been reported before [45, 57] with a critical distance of about 1 μm .

Considering the typical dimensions of extracellular space (volume fraction $\alpha = 0.2$, width 38 nm–64 nm [94]), one has to conclude that each point of interest will necessarily always be very close to another cell membrane under physiological conditions. Therefore, in reality Debye and diffusion layer effects will not be negligible. Instead, the effects of ion concentrations have to be taken into account. This indicates that representing the ES as a homogeneous medium by a single scalar resistivity ρ is not an accurate description.

Several previous models have dealt with this issue; in [20, 18], the authors used an inhomogeneous conductivity in the ES to account for the frequency filtering properties of extracellular space observed in experimental recordings. This approach might also be used to find an effective conductivity distribution that implicitly incorporates the concentration effects. However, it is unclear how to find the correct conductivities as a function of space. Even if the matching conductivities can be found, the question remains if such an approach will be successful in reproducing the full electrodiffusion dynamics and which spatial resolution will be required. The closest approximation to the electrodiffusion equations without explicitly resolving the Debye layer and including concentration effects only implicitly is described in [73]. A comparison to the full electrodiffusion model in a 3D setting would be of great interest.

In chapter 7, we made an attempt to quantify ephaptic effects between an axon fiber and a surrounding bundle of fibers. Due to the restrictions of the cylinder symmetry, this model does not represent the typical geometry of extracellular space, which is highly connected and tortuous, while it is rather isolated in our model. It is, however, not completely unrealistic either, as there is some evidence for regions with a very isolated ES or even “dead ends” in the extracellular matrix [94, e.g. Fig. 1]. In any case, the restricted volume and connectivity of the interstitial space in our model corresponds to a strongly reduced conductivity in terms of volume conductor theory. Such an increase in electric resistivity favors the development of large extracellular potentials. The simulation results show that already at relatively large inter-fiber distances of 500 nm, one axon can elicit an action potential in the surrounding fiber bundle purely by electrical coupling.

We also showed the effect of different intra- and extracellular volume ratios: while the membrane potential did not change appreciably, the relative distribution of potential contributions by the intra- or extracellular side changed to a large degree. This revealed another firing mode in which a large negative extracellular

potential could depolarize the cell, without a large change in the intracellular potential, as in the classical firing mode. This is plausible, since the ion channels only see the membrane potential difference $[\phi] = \phi_{\text{in}} - \phi_{\text{out}}$, therefore a depolarization can be reached by either increasing ϕ_{in} or by decreasing ϕ_{out} . This mode of firing is interestingly not an artifact of our model geometry: it has, in fact, been reported before in *in vitro* recordings of rat hippocampus CA1 cells (cf. section V.C in [40]) and the squid giant axon (originally in [86], and reviewed in [59, Fig. 1]).

Our model shows large extracellular potentials ranging up to 100 mV, e.g. in fig. 7.8d. The emerging extracellular potentials are sometimes even larger than the intracellular ones, due to the severely restricted volume. We also conducted some research on the experimental evidence for the large LFP magnitudes we found in chapter 7. To our surprise, comparable magnitudes have indeed been found in some cases. In [9], a number of references are given for *in vivo* measurements of large extracellular potentials with magnitudes larger than 50 mV.

It is clear that such large LFP magnitudes will suffice to elicit action potentials in neighboring cells, which is also the case in our model. When bringing the fibers closer together, we could even observe a “ping-pong effect”: the induced AP in the axon bundle in return produced a large extracellular response that was enough to elicit a second spike in the source axon. Again, we found evidence for such a phenomenon, mentioned in [9].

The authors also give a reason why such large potentials have only been measured in a few cases so far: they arise in areas where cells are most packed, i.e. where the conductivity is very low. These tightly packed areas are least accessible to electrodes, and any penetration by the electrode will inevitably increase the conductivity by destroying surrounding tissue, thereby diminishing the packing density and exposing the measurement site to regions with a higher extracellular volume. In [59], again the rodent CA1 region is mentioned in this context – together with a multitude of studies that assess the effect of electric fields within this area – since it has an extremely high resistivity (more than twice the intracellular one), presumably due to the very dense packing with an unusually low extracellular volume ratio of $\alpha = 0.12$.

One more word on the experimental point of view: if the (invasive) measurement of large LFPs is prevented by the measurement itself, this could mean that large extracellular potentials are a ubiquitous phenomenon that remains largely undetected with today’s experimental methods. If we take this thought further, one might also argue that many action potentials would not be identified as such in patch-clamp measurements, as only the intracellular potential ϕ_{in} with reference to a distant grounding electrode is observed. In extreme cases, action potentials

elicited by the second firing mode, i.e. triggered by a large negative extracellular potential ϕ_{out} without any large changes in the intracellular potential, would not be recognized. These “extracellularly-induced spikes” could only be seen by measuring both ϕ_{in} and ϕ_{out} , in order to obtain the true membrane potential $[[\phi]]$.

From a different perspective, the extracellular potentials generated by our model might also be of interest for another application. In [4], the authors mention the context of *parameter estimation* for neuron models, particularly estimating spatially distributed channel conductances from intracellular measurements made only at a small number of points – in the extreme case, from only one patch-clamp electrode. This problem is known to be an underdetermined optimization problem, given the fact that many different parameter sets can produce the same voltage response, as is shown in a plethora of different studies [1, 36, 37, 21, 95, 64, 101, 100, 99, 8].

The fact that there exist many local minima renders the application of classical, gradient-based methods extremely difficult, which is also the reason why many of the aforementioned studies utilized novel global optimization techniques like genetic algorithms or simulated annealing to overcome this problem – although there is also recent progress when using classical methods, as long as the range of stimuli given to the model is sufficiently diverse to tickle out the influence of every single parameter in this model [98, 66].

Given these difficulties in obtaining a meaningful and realistic parameter set for a given model, the authors in [4] suggest to include extracellular measurements next to the commonly used intracellular ones and report a markedly decreased variability in the obtained parameter sets. We note that using the presented electrodiffusion approach, extracellular signals may be used at any point in the [ES](#), while volume conductor models are restricted to certain minimum distances from the membrane, because they fail to reproduce the correct potentials close to the membrane. Of course, using electrodiffusion models for such parameter estimation procedures will increase the computational complexity to a large degree, but it might further increase the performance of the optimization strategy, as sharply varying (and therefore informative) juxtacellular signals may be used instead of the smoothed-out, less informative farfield signals.

In summary, we have shown that electrodiffusion models have proven useful in the context of neuroscience, especially when assessing the influence of variable concentration distributions on the extracellular electric potential. We hope that this thesis will be useful for readers who want to employ electrodiffusion models in comparable or related contexts.

The reader should keep in mind that the work presented here is just a first modeling attempt on this scale of detail, and the results open a number of further unanswered questions. Ongoing interdisciplinary research on this topic is needed to answer those. If this thesis helps stimulate follow-up investigations at the interface between scientific computing and neuroscience, our main objective has been achieved.

Bibliography

- [1] P. Achard and E. De Schutter. “Complex parameter landscape for a complex neuron model”. In: *PLoS Computational Biology* 2.7 (2006), e94. DOI: [10.1371/journal.pcbi.0020094](https://doi.org/10.1371/journal.pcbi.0020094) (cit. on p. 146).
- [2] A. Agudelo-Toro. “Numerical Simulations on the Biophysical Foundations of the Neuronal Extracellular Space”. PhD thesis. Göttingen, Georg-August Universität, Diss., 2012, 2013 (cit. on pp. 8, 37).
- [3] D. J. Aidley. *The physiology of excitable cells*. Cambridge University Press, 1998. ISBN: 9780521574211 (cit. on p. 126).
- [4] C. A. Anastassiou, C. Buzsaki, C. Koch, R. Quiroga, and S. Panzeri. “Biophysics of extracellular spikes”. In: *Principles of Neural Coding* 15 (2013) (cit. on p. 146).
- [5] C. A. Anastassiou, R. Perin, H. Markram, and C. Koch. “Ephaptic coupling of cortical neurons”. In: *Nature neuroscience* 14.2 (2011), pp. 217–223. DOI: [10.1038/nn.2727](https://doi.org/10.1038/nn.2727) (cit. on p. 126).
- [6] A. Arvanitaki. “Effects evoked in an axon by the activity of a contiguous one”. In: *Journal of neurophysiology* (1942). Eprint: <http://jn.physiology.org/content/5/2/89.full.pdf> (cit. on p. 126).
- [7] F. Bähner, E. K. Weiss, G. Birke, N. Maier, D. Schmitz, U. Rudolph, M. Frotscher, R. D. Traub, M. Both, and A. Draguhn. “Cellular correlate of assembly formation in oscillating hippocampal networks in vitro”. In: *Proceedings of the National Academy of Sciences* 108.35 (2011), E607–E616. DOI: [10.1073/pnas.1103546108](https://doi.org/10.1073/pnas.1103546108) (cit. on p. 25).
- [8] J. R. Banga, C. G. Moles, and A. A. Alonso. “Global optimization of bioprocesses using stochastic and hybrid methods”. In: *Frontiers in global optimization*. Springer, 2004, pp. 45–70. DOI: [10.1007/978-1-4613-0251-3_3](https://doi.org/10.1007/978-1-4613-0251-3_3) (cit. on p. 146).
- [9] R. C. Barr and R. Plonsey. “Electrophysiological interaction through the interstitial space between adjacent unmyelinated parallel fibers”. In: *Biophysical journal* 61.5 (1992), pp. 1164–1175. DOI: [10.1016/S0006-3495\(92\)81925-2](https://doi.org/10.1016/S0006-3495(92)81925-2) (cit. on pp. 135, 145).

-
- [10] P. Bastian. “Parallele adaptive Mehrgitterverfahren”. ICA–Bericht 94–1. PhD thesis. Universität Heidelberg, 1994. URL: http://conan.iwr.uni-heidelberg.de/people/peter/pdf/Bastian_phdthesis_2004.pdf (cit. on p. 24).
- [11] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander. “A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and tests in DUNE”. In: *Computing* 82.2–3 (2008), pp. 121–138. DOI: [10.1007/s00607-008-0004-9](https://doi.org/10.1007/s00607-008-0004-9) (cit. on p. 58).
- [12] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander. “A generic grid interface for parallel and adaptive scientific computing. Part I: abstract framework”. In: *Computing* 82.2–3 (2008), pp. 103–119. DOI: [10.1007/s00607-008-0003-x](https://doi.org/10.1007/s00607-008-0003-x) (cit. on p. 58).
- [13] P. Bastian, M. Blatt, A. Dedner, C. Engwer, J. Fahlke, C. Gräser, R. Klöfkorn, M. Nolte, M. Ohlberger, and O. Sander. *DUNE Web page*. URL: <http://www.dune-project.org> (visited on 12/04/2012) (cit. on p. 58).
- [14] P. Bastian. *Lecture Notes on Scientific Computing with Partial Differential Equations*. 2012. URL: http://conan.iwr.uni-heidelberg.de/teaching/numerik2_ss2014/num2.pdf (visited on 04/17/2014) (cit. on pp. 11, 12, 17, 18, 22).
- [15] P. Bastian, F. Heimann, and S. Marnach. “Generic implementation of finite element methods in the Distributed and Unified Numerics Environment (DUNE)”. In: *Kybernetika* 46.2 (2010), pp. 294–315. URL: <http://dml.cz/dmlcz/140745> (cit. on p. 59).
- [16] S. B. Baumann, D. R. Wozny, S. K. Kelly, and F. M. Meno. “The electrical conductivity of human cerebrospinal fluid at body temperature”. In: *IEEE Trans. Biomed. Eng.* 44.3 (1997), pp. 220–223. DOI: [10.1109/10.554770](https://doi.org/10.1109/10.554770) (cit. on p. 93).
- [17] M. F. Bear, B. W. Connors, and M. A. Paradiso. *Neuroscience*. Vol. 2. Lippincott Williams & Wilkins, 2007. ISBN: 9780781760034 (cit. on p. 30).
- [18] C. Bédard and A. Destexhe. “Macroscopic Models of Local Field Potentials and the Apparent 1/f Noise in Brain Activity”. In: *Biophys. J.* 96.7 (2009), pp. 2589–2603. DOI: [10.1016/j.bpj.2008.12.3951](https://doi.org/10.1016/j.bpj.2008.12.3951) (cit. on pp. 8, 144).
- [19] C. Bédard, H. Kröger, and A. Destexhe. “Model of low-pass filtering of local field potentials in brain tissue”. In: *Phys. Rev. E* 73.5 (2006), p. 051911. DOI: [10.1103/PhysRevE.73.051911](https://doi.org/10.1103/PhysRevE.73.051911). URL: <http://link.aps.org/doi/10.1103/PhysRevE.73.051911> (cit. on p. 8).

- [20] C. Bédard, H. Kröger, and A. Destexhe. “Modeling extracellular field potentials and the frequency-filtering properties of extracellular space”. In: *Biophys. J.* 86.3 (2004), pp. 1829–1842. DOI: [10.1016/S0006-3495\(04\)74250-2](https://doi.org/10.1016/S0006-3495(04)74250-2) (cit. on pp. 8, 144).
- [21] U. S. Bhalla and J. M. Bower. “Exploring parameter space in detailed single neuron models: simulations of the mitral and granule cells of the olfactory bulb”. In: *Journal of Neurophysiology* 69.6 (1993), pp. 1948–1965. Eprint: <http://jn.physiology.org/content/69/6/1948.full.pdf> (cit. on p. 146).
- [22] P. Biler, W. Hebisch, and T. Nadzieja. “The Debye system: existence and large time behavior of solutions”. In: *Nonlinear Analysis: Theory, Methods & Applications* 23.9 (1994), pp. 1189–1209. DOI: [10.1016/0362-546X\(94\)90101-5](https://doi.org/10.1016/0362-546X(94)90101-5) (cit. on p. 39).
- [23] J. Bower, D. Beeman, and A. Wylde. *The book of GENESIS: exploring realistic neural models with the GEneral NEural Simulation System*. Telos New York, 1995. ISBN: 0387940197 (cit. on p. 6).
- [24] S. C. Brenner and R. Scott. *The mathematical theory of finite element methods*. Vol. 15. Springer, 2008. ISBN: 9780387759333 (cit. on p. 18).
- [25] C. Brezinski and M. Redivo-Zaglia. “Look-ahead in Bi-CGSTAB and other product methods for linear systems”. In: *BIT Numerical Mathematics* 35.2 (1995), pp. 169–201. DOI: [10.1007/BF01737161](https://doi.org/10.1007/BF01737161) (cit. on p. 109).
- [26] G. Buzsáki, C. A. Anastassiou, and C. Koch. “The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes”. In: *Nat. Rev. Neurosci.* 13.6 (June 2012), pp. 407–420. ISSN: 1471-003X. DOI: [10.1038/nrn3241](https://doi.org/10.1038/nrn3241). URL: http://www.nature.com/nrn/journal/v13/n6/supinfo/nrn3241_S1.html (cit. on p. 142).
- [27] N. Carnevale. *Vtrap derived*. Oct. 2007. URL: <http://www.neuron.yale.edu/phpBB/viewtopic.php?f=15&t=1075> (visited on 12/04/2012) (cit. on p. 34).
- [28] D. Chapman. “Diffuse Distribution of Adsorbed Ions”. In: *Phil. Mag.* 25 (1913), p. 475 (cit. on p. 27).
- [29] J. Clark and R. Plonsey. “A mathematical evaluation of the core conductor model”. In: *Biophysical journal* 6.1 (1966), pp. 95–112. DOI: [10.1016/S0006-3495\(66\)86642-0](https://doi.org/10.1016/S0006-3495(66)86642-0) (cit. on p. 129).

-
- [30] R. D. Coalson and M. G. Kurnikova. “Poisson-Nernst-Planck theory approach to the calculation of current through biological ion channels”. In: *NanoBioscience, IEEE Transactions on* 4.1 (2005), pp. 81–93. DOI: [10.1109/TNB.2004.842495](https://doi.org/10.1109/TNB.2004.842495) (cit. on p. 9).
- [31] K. Cole. *Membranes, Ions, and Impulses: A Chapter of Classical Biophysics*. Biophysics series v. 5. University of California Press, 1968. ISBN: 9780520002517 (cit. on p. 80).
- [32] P. Debye and E. Huckel. “On the theory of electrolytes. I. Freezing point depression and related phenomena”. In: *The Collected Papers of Peter JW Debye* (1954), pp. 217–263 (cit. on p. 28).
- [33] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. “A supernodal approach to sparse partial pivoting”. In: *SIAM J. Matrix Anal. Appl.* 20.3 (1999), pp. 720–755. DOI: [10.1137/S0895479895291765](https://doi.org/10.1137/S0895479895291765) (cit. on p. 52).
- [34] P. Deuffhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics. Springer, 2011. ISBN: 9783642238987 (cit. on p. 22).
- [35] D. A. Drachman. “Do we have brain to spare?” In: *Neurology* 64.12 (2005), pp. 2004–2005. DOI: [10.1212/01.WNL.0000166914.38327.BB](https://doi.org/10.1212/01.WNL.0000166914.38327.BB) (cit. on p. 2).
- [36] S. Druckmann, Y. Banitt, A. A. Gidon, F. Schürmann, H. Markram, and I. Segev. “A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data”. In: *Frontiers in neuroscience* 1 (2007), p. 1. DOI: [10.3389/neuro.01.1.1.001.2007](https://doi.org/10.3389/neuro.01.1.1.001.2007) (cit. on p. 146).
- [37] S. Druckmann, T. K. Berger, S. Hill, F. Schürmann, H. Markram, and I. Segev. “Evaluating automated parameter constraining procedures of neuron models by experimental and surrogate data”. In: *Biological cybernetics* 99.4-5 (2008), pp. 371–379. DOI: [10.1007/s00422-008-0269-2](https://doi.org/10.1007/s00422-008-0269-2) (cit. on p. 146).
- [38] B. Eisenberg and W. Liu. “Poisson-Nernst-Planck systems for ion channels with permanent charges”. In: *SIAM Journal on Mathematical Analysis* 38.6 (2007), pp. 1932–1966. DOI: [10.1137/060657480](https://doi.org/10.1137/060657480) (cit. on p. 39).
- [39] K. Eriksson. *Computational differential equations*. Vol. 1. Cambridge University Press, 1996. ISBN: 9780521567381 (cit. on pp. 16, 21).

- [40] D. S. Faber and H. Korn. “Electrical field effects: their relevance in central neural networks”. In: *Physiological reviews* 69.3 (1989), pp. 821–863. Eprint: <http://physrev.physiology.org/content/69/3/821.full.pdf> (cit. on pp. 9, 93, 126, 145).
- [41] B. Frankenhaeuser and A. Hodgkin. “The after-effects of impulses in the giant nerve fibres of *Loligo*”. In: *The Journal of physiology* 131.2 (1956), pp. 341–376 (cit. on pp. 97, 98).
- [42] S. Gabriel, R. Lau, and C. Gabriel. “The dielectric properties of biological tissues: III. Parametric models for the dielectric spectrum of tissues”. In: *Phys. Med. Biol.* 41.11 (1996), p. 2271. DOI: [10.1088/0031-9155/41/11/003](https://doi.org/10.1088/0031-9155/41/11/003) (cit. on p. 93).
- [43] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994. ISBN: 9780321700698 (cit. on pp. 62, 64).
- [44] *Git*. Nov. 2013. URL: <http://www.git-scm.com> (visited on 11/06/2013) (cit. on p. 60).
- [45] C. Gold, D. Henze, C. Koch, and G. Buzsáki. “On the origin of the extracellular action potential waveform: a modeling study”. In: *J. Neurophysiol.* 95.5 (2006), pp. 3113–3128. DOI: [10.1152/jn.00979.2005](https://doi.org/10.1152/jn.00979.2005) (cit. on pp. 8, 46, 144).
- [46] G. Gouy. “On the Formation of Electrical Charges at the Surface of an Electrolyte”. In: *J. physique* 9 (1910), pp. 457–469 (cit. on p. 27).
- [47] D. E. Gutstein, G. E. Morley, H. Tamaddon, D. Vaidya, M. D. Schneider, J. Chen, K. R. Chien, H. Stuhlmann, and G. I. Fishman. “Conduction slowing and sudden arrhythmic death in mice with cardiac-restricted inactivation of connexin43”. In: *Circulation research* 88.3 (2001), pp. 333–339. DOI: [10.1161/01.RES.88.3.333](https://doi.org/10.1161/01.RES.88.3.333) (cit. on p. 126).
- [48] A. C. Guyton and J. Hall. *Textbook of medical physiology*. Philadelphia, Pa: Saunders/Elsevier, 2011. ISBN: 978-1416045748 (cit. on p. 126).
- [49] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics. Springer, 2011. ISBN: 9783540566700 (cit. on pp. 11, 23).
- [50] G. Hairer. *Solving Ordinary Differential Equations II*. Springer Berlin Heidelberg, 2010. ISBN: 9783642052224 (cit. on pp. 11, 23).

-
- [51] H. Helmholtz. “Über einige Gesetze der Vertheilung elektrischer Ströme in körperlichen Leitern mit Anwendung auf die thierisch-elektrischen Versuche”. In: *Annalen der Physik* 165.6 (1853), pp. 211–233. ISSN: 1521-3889. DOI: [10.1002/andp.18531650603](https://doi.org/10.1002/andp.18531650603) (cit. on p. 27).
- [52] B. Hille. *Ion channels of excitable membranes*. 3. ed. Sunderland, Mass.: Sinauer Assoc., 2001. ISBN: 978-0-87893-321-1 (cit. on p. 31).
- [53] M. Hines and N. Carnevale. “The NEURON simulation environment”. In: *Neural Comput.* 9.6 (1997), pp. 1179–1209. DOI: [10.1162/neco.1997.9.6.1179](https://doi.org/10.1162/neco.1997.9.6.1179) (cit. on pp. 4, 6, 36, 77).
- [54] R. K. Hobbie and B. J. Roth. *Intermediate physics for medicine and biology*. Springer, 2007. ISBN: 978-0-387-30942-2 (cit. on p. 102).
- [55] A. Hodgkin and A. Huxley. “A quantitative description of membrane current and its application to conduction and excitation in nerve”. In: *J. Physiol.* 117.4 (1952), p. 500 (cit. on pp. 6, 32, 69).
- [56] G. R. Holt. “A critical reexamination of some assumptions and implications of cable theory in neurobiology”. PhD thesis. California Institute of Technology, 1997 (cit. on pp. 6, 8, 127, 129).
- [57] G. R. Holt and C. Koch. “Electrical interactions via the extracellular potential near cell bodies”. In: *J. Comput. Neurosci.* 6.2 (1999), pp. 169–184. DOI: [10.1023/A:1008832702585](https://doi.org/10.1023/A:1008832702585) (cit. on pp. 37, 144).
- [58] T. R. Insel, S. C. Landis, and F. S. Collins. “The NIH brain initiative”. In: *Science* 340.6133 (2013), pp. 687–688. DOI: [10.1126/science.1239276](https://doi.org/10.1126/science.1239276) (cit. on p. 3).
- [59] J. Jefferys. “Nonsynaptic modulation of neuronal activity in the brain: electric currents and extracellular ions”. In: *Physiological reviews* 75.4 (1995), pp. 689–723. Eprint: <http://physrev.physiology.org/content/75/4/689.full.pdf> (cit. on pp. 9, 126, 145).
- [60] J. W. Jerome and T. Kerkhoven. “A finite element approximation theory for the drift diffusion semiconductor model”. In: *SIAM journal on numerical analysis* 28.2 (1991), pp. 403–422. DOI: [10.1137/0728023](https://doi.org/10.1137/0728023) (cit. on p. 39).
- [61] N. Josuttis. *The C++ Standard Library: A Tutorial and Reference*. C++ programming languages. Addison-Wesley, 1999. ISBN: 9780201379266 (cit. on p. 61).
- [62] B. Katz and O. H. Schmitt. “Electric interaction between two adjacent nerve fibres”. In: *The Journal of physiology* 97.4 (1940), pp. 471–488 (cit. on p. 126).

- [63] D. Kempf. *Numerische Simulation einer Geothermieanlage*. Apr. 2011. URL: http://conan.iwr.uni-heidelberg.de/teaching/softwarepraktika/Bericht_Dominic_Kempf.pdf (visited on 11/06/2013) (cit. on p. 61).
- [64] N. Keren, N. Peled, and A. Korngreen. “Constraining compartmental models using multiple voltage recordings and genetic algorithms”. In: *Journal of neurophysiology* 94.6 (2005), pp. 3730–3742. DOI: [10.1152/jn.00408.2005](https://doi.org/10.1152/jn.00408.2005) (cit. on p. 146).
- [65] C. Koch. *Biophysics of computation: information processing in single neurons*. Oxford University Press, USA, 2004. ISBN: 9780195181999 (cit. on pp. 32, 77).
- [66] M. Kostuk, B. Toth, C. D. Meliza, D. Margoliash, and H. D. Abarbanel. “Dynamical estimation of neuron and network properties II: path integral Monte Carlo methods”. In: *Biological Cybernetics* 106.3 (2012), pp. 155–167. ISSN: 0340-1200. DOI: [10.1007/s00422-012-0487-5](https://doi.org/10.1007/s00422-012-0487-5) (cit. on p. 146).
- [67] R. Lipowsky and E. Sackmann. *Structure and Dynamics of Membranes: I. From Cells to Vesicles/II. Generic and Specific Interactions*. Vol. 1. Access Online via Elsevier, 1995. ISBN: 9780080541914 (cit. on pp. 27, 28, 39, 85).
- [68] C. Loppreore, T. Bartol, J. Coggan, D. Keller, G. Sosinsky, M. Ellisman, and T. Sejnowski. “Computational modeling of three-dimensional electrodiffusion in biological systems: Application to the node of ranvier”. In: *Biophys. J.* 95.6 (2008), pp. 2624–2635. DOI: [10.1529/biophysj.108.132167](https://doi.org/10.1529/biophysj.108.132167) (cit. on p. 9).
- [69] B. Lu, M. J. Holst, J. Andrew McCammon, and Y. Zhou. “Poisson–Nernst–Planck equations for simulating biomolecular diffusion–reaction processes I: Finite element solutions”. In: *Journal of computational physics* 229.19 (2010), pp. 6979–6994. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2010.05.035](https://doi.org/10.1016/j.jcp.2010.05.035) (cit. on pp. 9, 39, 69).
- [70] H. Markram. “The blue brain project”. In: *Nature Reviews Neuroscience* 7.2 (2006), pp. 153–160. DOI: [10.1038/nrn1848](https://doi.org/10.1038/nrn1848) (cit. on p. 4).
- [71] H. Markram. “The human brain project”. In: *Scientific American* 306.6 (2012), pp. 50–55. DOI: [10.1038/scientificamerican0612-50](https://doi.org/10.1038/scientificamerican0612-50) (cit. on p. 3).
- [72] Y. Min, K. Kristiansen, J. M. Boggs, C. Husted, J. A. Zasadzinski, and J. Israelachvili. “Interaction forces and adhesion of supported myelin lipid bilayers modulated by myelin basic protein”. In: *Proceedings of the National Academy of Sciences* 106.9 (2009), pp. 3154–3159. DOI: [10.1073/pnas.0813110106](https://doi.org/10.1073/pnas.0813110106) (cit. on p. 102).

- [73] Y. Mori. “A three-dimensional model of cellular electrical activity”. PhD thesis. New York University, 2006 (cit. on pp. 9, 97, 101, 127, 130, 143, 144).
- [74] Y. Mori. *From three-dimensional electrophysiology to the cable model: an asymptotic study*. 2009. arXiv: 0901.3914 [q-bio.NC] (cit. on p. 8).
- [75] Y. Mori, G. I. Fishman, and C. S. Peskin. “Ephaptic conduction in a cardiac strand model with 3D electrodiffusion”. In: *Proceedings of the National Academy of Sciences* 105.17 (2008), pp. 6463–6468. DOI: 10.1073/pnas.0801089105 (cit. on pp. 9, 127, 143).
- [76] S. Müthing. *dune-multidomain*. URL: <http://users.dune-project.org/projects/dune-multidomain> (visited on 12/04/2012) (cit. on p. 59).
- [77] S. Müthing and P. Bastian. “Dune-multidomaingrid: A metagrid approach to subdomain modeling”. In: *Advances in DUNE*. Ed. by A. Dedner, B. Flemisch, and R. Klöfkorn. Springer Berlin Heidelberg, 2012, pp. 59–73. ISBN: 978-3-642-28588-2. DOI: 10.1007/978-3-642-28589-9_5 (cit. on p. 59).
- [78] R. B. Northrop. *Introduction to dynamic modeling of neuro-sensory systems*. CRC Press, 2000. ISBN: 9780849308147 (cit. on p. 102).
- [79] P. Nunez and R. Srinivasan. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2005. ISBN: 9780195050387 (cit. on p. 142).
- [80] K. Pettersen, E. Hagen, and G. Einevoll. “Estimation of population firing rates and current source densities from laminar electrode recordings”. In: *J. Comput. Neurosci.* 24.3 (2008), pp. 291–313. DOI: 10.1007/s10827-007-0056-4 (cit. on p. 8).
- [81] R. Plonsey. *Volume conductor theory*. 1995 (cit. on pp. 7, 36).
- [82] J. Pods, J. Schönke, and P. Bastian. “Electrodiffusion Models of Neurons and Extracellular Space Using the Poisson-Nernst-Planck Equations – Numerical Simulation of the Intra-and Extracellular Potential for an Axon Model”. In: *Biophysical journal* 105.1 (2013), pp. 242–254. DOI: 10.1016/j.bpj.2013.05.041 (cit. on pp. 10, 41, 69, 93).
- [83] N. Qian and T. Sejnowski. “An electro-diffusion model for computing membrane potentials and ionic concentrations in branching dendrites, spines and axons”. In: *Biological Cybernetics* 62.1 (1989), pp. 1–15. DOI: 10.1007/BF00217656 (cit. on pp. 7, 8).

- [84] W. Rall. “Cable theory for dendritic neurons”. In: *Methods in neuronal modeling*. MIT press, 1989, pp. 9–92. ISBN: 0-262-11133-0 (cit. on pp. 6, 35).
- [85] W. Rall. “Core conductor theory and cable properties of neurons”. In: *Comprehensive Physiology* (2011). DOI: [10.1002/cphy.cp010103](https://doi.org/10.1002/cphy.cp010103) (cit. on p. 129).
- [86] F. Ramon and J. W. Moore. “Ephaptic transmission in squid giant axons”. In: *American Journal of Physiology-Cell Physiology* 234.5 (1978), pp. C162–C169. Eprint: <http://ajpcell.physiology.org/content/234/5/C162.full.pdf> (cit. on p. 145).
- [87] P. Rosenfalck. “Intra-and extracellular potential fields of active nerve and muscle fibres. A physico-mathematical analysis of different models.” In: *Thrombosis et diathesis haemorrhagica, Supplementum* 321 (1968), pp. 1–168 (cit. on p. 8).
- [88] Y. Saad. *Iterative methods for sparse linear systems*. Siam, 2003. ISBN: 9780898718003 (cit. on p. 24).
- [89] J. Schönke. “Unsteady analytical solutions to the Poisson-Nernst-Planck equations”. In: *Journal of Physics A: Mathematical and Theoretical* 45.45 (2012), p. 455204. DOI: [10.1088/1751-8113/45/45/455204](https://doi.org/10.1088/1751-8113/45/45/455204). URL: <http://stacks.iop.org/1751-8121/45/i=45/a=455204> (cit. on pp. 52, 57).
- [90] V. Simoncini and D. B. Szyld. “Recent computational developments in Krylov subspace methods for linear systems”. In: *Numerical Linear Algebra with Applications* 14.1 (2007), pp. 1–59. DOI: [10.1002/nla.499](https://doi.org/10.1002/nla.499) (cit. on p. 109).
- [91] N. Sperlakis and J. E. Mann Jr. “Evaluation of electric field changes in the cleft between excitable cells”. In: *Journal of theoretical biology* 64.1 (1977), pp. 71–96. DOI: [10.1016/0022-5193\(77\)90114-X](https://doi.org/10.1016/0022-5193(77)90114-X) (cit. on p. 126).
- [92] K. Stephenson. *Introduction to circle packing: The theory of discrete analytic functions*. Cambridge University Press, 2005. ISBN: 978-0521823562 (cit. on p. 130).
- [93] O. Stern. “Theory of the electrical double layer (in German)”. In: *Electrochemistry* 30 (1924), pp. 508–516 (cit. on p. 27).
- [94] E. Syková and C. Nicholson. “Diffusion in brain extracellular space”. In: *Physiological reviews* 88.4 (2008), p. 1277. DOI: [10.1152/physrev.00027.2007](https://doi.org/10.1152/physrev.00027.2007) (cit. on pp. 129, 130, 144).

-
- [95] J. Tabak, C. R. Murphey, and L. Moore. “Parameter estimation methods for single neuron models”. In: *Journal of computational neuroscience* 9.3 (2000), pp. 215–236. DOI: [10.1023/A:1026531603628](https://doi.org/10.1023/A:1026531603628) (cit. on p. 146).
- [96] The Apache Software Foundation. *Apache Subversion*. URL: <http://subversion.apache.org> (visited on 11/06/2013) (cit. on p. 60).
- [97] The HDF Group. *Hierarchical data format version 5*. 2000–2010. URL: <http://www.hdfgroup.org/HDF5> (visited on 11/06/2013) (cit. on p. 68).
- [98] B. A. Toth, M. Kostuk, C. D. Meliza, D. Margoliash, and H. D. Abarbanel. “Dynamical estimation of neuron and network properties I: variational methods”. In: *Biological cybernetics* 105.3-4 (2011), pp. 217–237. DOI: [10.1007/s00422-011-0459-1](https://doi.org/10.1007/s00422-011-0459-1) (cit. on p. 146).
- [99] W. Van Geit, E. De Schutter, and P. Achard. “Automated neuron model optimization techniques: a review”. In: *Biological cybernetics* 99.4-5 (2008), pp. 241–251. DOI: [10.1007/s00422-008-0257-6](https://doi.org/10.1007/s00422-008-0257-6) (cit. on p. 146).
- [100] W. Van Geit, P. Achard, and E. De Schutter. “Neurofitter: a parameter tuning package for a wide range of electrophysiological neuron models”. In: *BMC Neuroscience* 8.Suppl 2 (2007), P5. DOI: [10.1186/1471-2202-8-S2-P5](https://doi.org/10.1186/1471-2202-8-S2-P5) (cit. on p. 146).
- [101] M. C. Vanier and J. M. Bower. “A comparative survey of automated parameter-search methods for compartmental neural models”. In: *Journal of computational neuroscience* 7.2 (1999), pp. 149–171. DOI: [10.1023/A:1008972005316](https://doi.org/10.1023/A:1008972005316) (cit. on p. 146).
- [102] T. Veldhuizen. “Techniques for Scientific C++”. In: (2000) (cit. on pp. 58, 63, 68).
- [103] S. A. Weiss and D. S. Faber. “Field effects in the CNS play functional roles”. In: *Frontiers in neural circuits* 4 (2010), p. 15. DOI: [10.3389/fncir.2010.00015](https://doi.org/10.3389/fncir.2010.00015) (cit. on p. 126).
- [104] T. Williams, C. Kelley, and many others. *Gnuplot 4.6: an interactive plotting program*. URL: <http://gnuplot.sourceforge.net> (visited on 11/06/2013) (cit. on p. 68).
- [105] H. Wu and J. Jiang. “Global solution to the drift-diffusion-Poisson system for semiconductors with nonlinear recombination-generation rate.” In: *Asymptotic Analysis* 85.1–2 (2013), pp. 75–105. DOI: [10.3233/ASY-131176](https://doi.org/10.3233/ASY-131176). URL: <http://dblp.uni-trier.de/db/journals/asy/asy85.html#WuJ13> (cit. on p. 39).

Bibliography

- [106] B. Wyttenbach. *Supplement: Electrical Characteristics of Neurons*. Oct. 2008. URL: <http://132.236.112.18/wytt/1/supplement1.pdf> (visited on 04/14/2014) (cit. on p. 31).

A Results of the Volume-Corrected Axon Bundle Simulations

The detailed results of the volume-corrected axon bundle simulations are shown in figs. A.1 to A.4. Table A.1 shows the statistics for different inter-fiber distances d . Note that only the setup $d = 50$ nm was compiled with full optimizations, the other cases were accidentally computed without compiler optimizations, resulting in a factor of 6-7 in the total computation time.

Table A.1: **Simulation timings for the axon bundle setup using different inter-fiber distances d and a volume-corrected bundle diameter d_b .** For each distance d , the number of unknowns, total computation time, the needed number of time steps and the average solution time per time step (full Newton iteration) is shown.

d [nm]	# DOFs	Total comp. time [s]	# time steps	avg. time / time step
1000	116352	157710	2027	77.8
500	114332	173738	2027	85.71
100	114332	183747	2027	90.65
50	114332	26027.7	2026	12.85

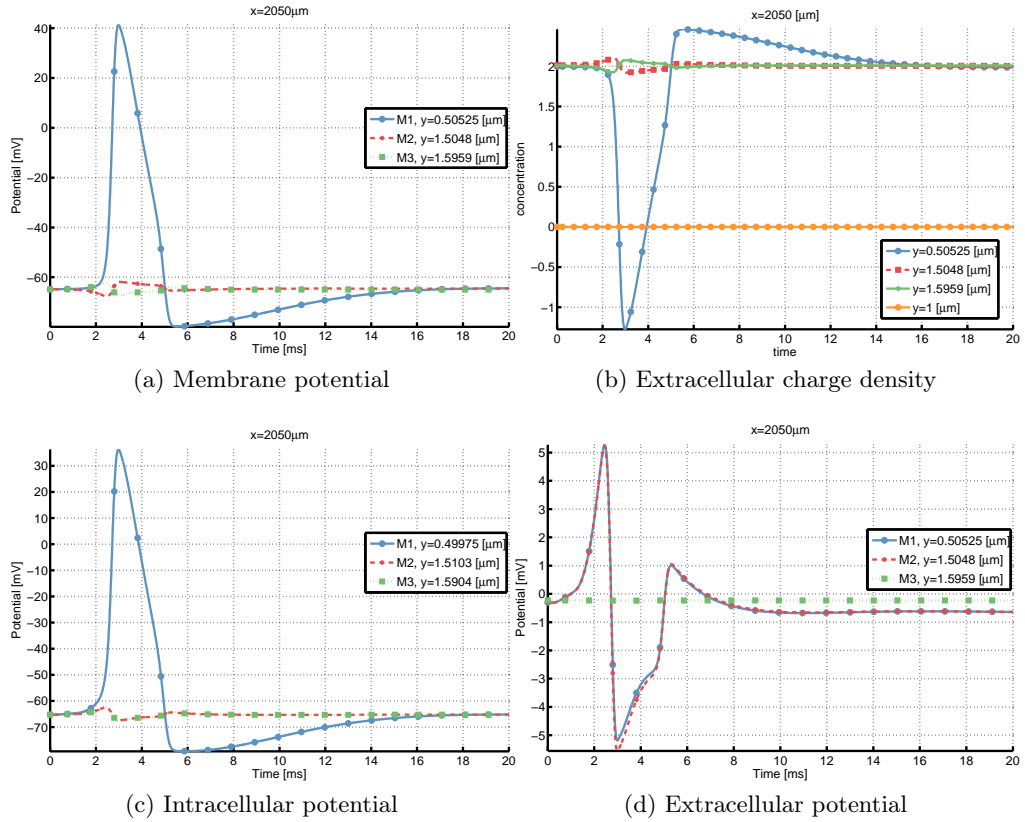


Figure A.1: Potential and charge density time courses for $d = 1 \mu\text{m}$ (volume-corrected).

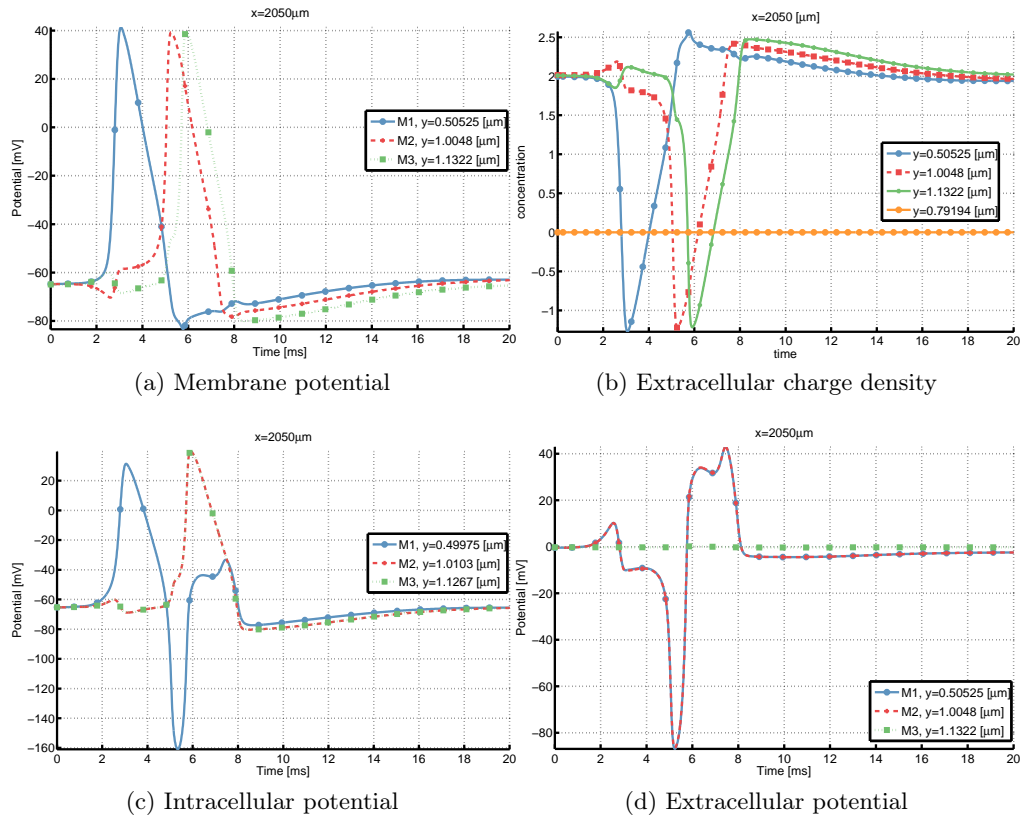


Figure A.2: Potential and charge density time courses for $d = 500$ nm (volume-corrected).

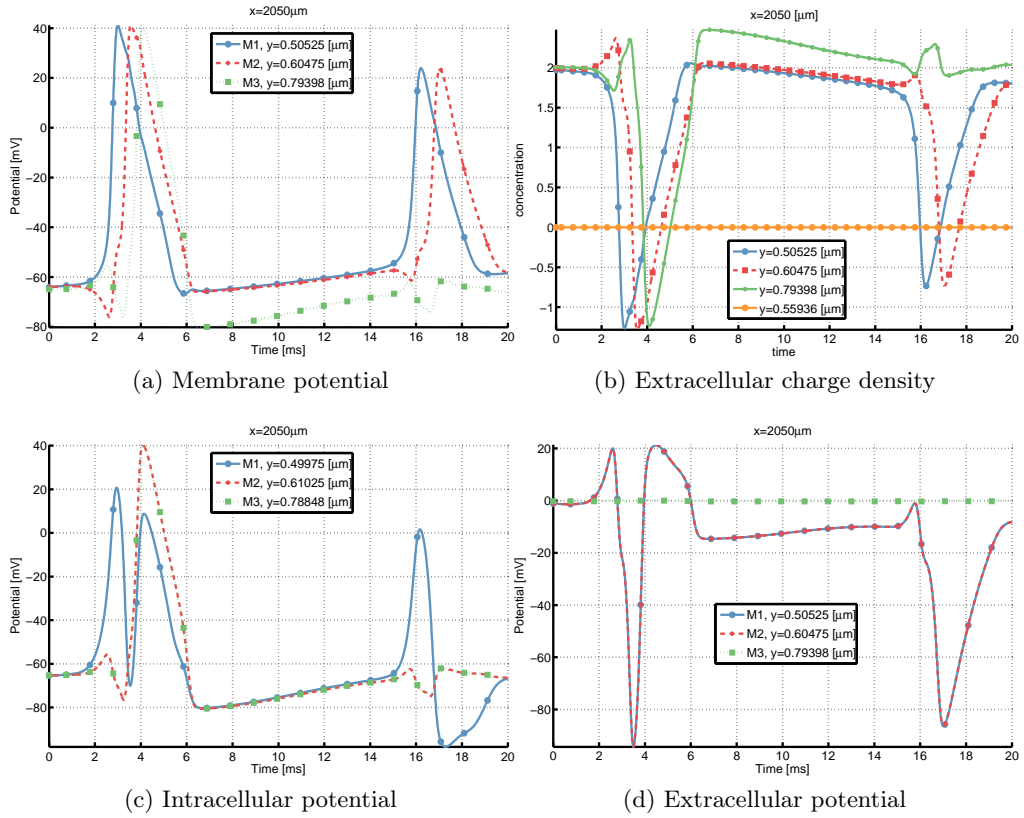


Figure A.3: Potential and charge density time courses for $d = 100$ nm (volume-corrected).

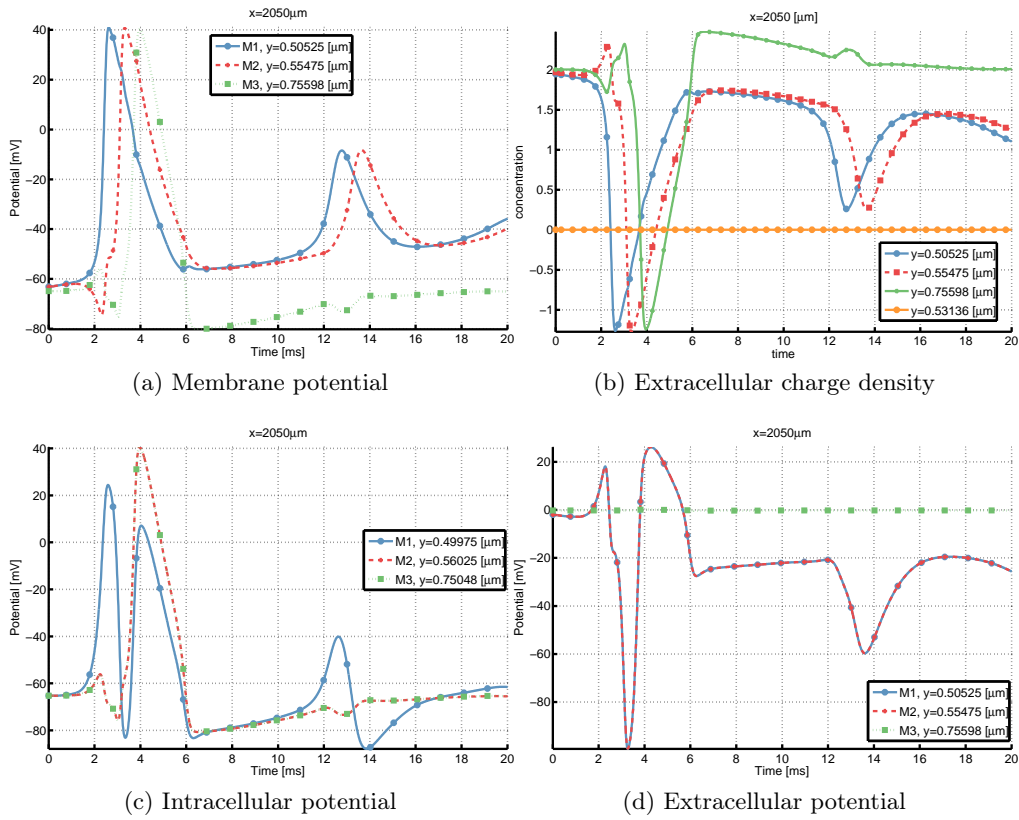


Figure A.4: Potential and charge density time courses for $d = 50$ nm (volume-corrected).

Acronyms

- AMG** algebraic multigrid. 52, 67, 109
- AP** action potential. 3, 24, 36, 47, 51, 71, 74, 75, 78, 79, 83, 87, 88, 91, 94, 96, 104, 108, 110–112, 114–120, 123, 126, 127, 132–134, 136, 137, 140, 143, 145
- BiCGStab** stabilized biconjugate gradient. 52, 67, 72, 73, 109
- cG** continuous Galerkin. 19
- dG** discontinuous Galerkin. 19
- DOF** degree of freedom. 4, 5, 66, 67, 70–72, 103, 107, 110, 132, 141, 159
- EAP** extracellular action potential. 36, 80, 82–84, 87, 90, 94, 96–98, 101, 118–120, 123, 125, 135–137
- ED** electrodiffusion. 94–97, 100, 124
- EDL** electric double layer. 26–28, 97
- EEG** electroencephalography. 3, 142
- ES** extracellular space. 24, 41, 131, 134, 144, 146
- FEM** finite element method. 9, 11, 15, 18, 22, 23, 48
- GHK** Goldman-Hodgkin-Katz. 30
- GMRes** generalized minimal residual. 52, 67, 109, 112
- HH** Hodgkin-Huxley. 34–36, 40, 41, 46, 47, 51, 52, 61, 69, 79, 80, 101, 134
- HPC** high performance computing. 4, 72
- ILU** inexact LU. 52, 67, 72, 73, 109, 141

- IO** input/output. 5
- ISTL** iterative solver template library. 59, 67
- LFP** local field potential. 3, 8, 36, 123, 126, 142, 145
- LS** linear solver. 72, 73, 112
- LSA** line source approximation. 8, 38, 46, 93–99, 101, 113, 120, 122–125, 127
- ODE** ordinary differential equation. 11, 15, 23, 34, 51
- PCM** parallel conductance model. 31, 46, 47, 75, 77
- PDE** partial differential equation. 5, 6, 11–17, 19, 20, 22–24, 35, 39, 58, 98, 141
- PNP** Poisson-Nernst-Planck. 8, 9, 14, 39, 41, 44, 47, 48, 52, 53, 68, 73, 79, 91, 97, 98, 101, 141, 142
- TMP** template meta program. 63, 68
- VC** volume conductor. 7, 9, 10, 98, 100

Danksagung

Geld ist (Arbeits-)Zeit, und daher danke ich zuallererst dem Bundesministerium für Bildung und Forschung für die Schaffung des BCCN Heidelberg/Mannheim und die damit verbundene Finanzierung meiner Arbeit (BMBF No. 01GQ1003A).

Weiterhin möchte ich Prof. Peter Bastian dafür danken, dass er mir die Promotion zu diesem hochinteressanten Thema ermöglicht hat. Die parallele Einarbeitung in mehrere neue Themengebiete war herausfordernd und mit vielen Hindernissen verbunden. Für die Hilfestellung besonders im Bereich der numerischen Mathematik möchte ich mich ebenso bedanken wie für die angenehme Arbeitsatmosphäre, zu denen auch alle anderen Mitglieder der AG Paralleles Rechnen beigetragen haben. Allen Kollegen gilt daher mein herzlicher Dank für die vergangenen 3 1/2 Jahre.

Dem IWR und der Fakultät für Mathematik und Informatik der Universität Heidelberg möchte für die bereitgestellte Infrastruktur und die Möglichkeit einer interdisziplinären Promotion an der Schnittstelle zwischen Informatik, Mathematik und Biophysik danken.

Besonderer Dank geht an Dan Popović, der mich zu Beginn in die Untiefen von DUNE und der Heidelberger Kneipenkultur eingeführt hat. Viele Gespräche mit oder ohne Fußballbezug haben geholfen, die schwierige Anfangszeit zu meistern. Großer Dank gilt auch Johannes Schönke bzw. Prof. Henry, der mich thematisch in die richtige Spur gebracht hat, indem er mich zuerst auf den Beifahrersitz des Elektrodifusionswagens gehievt hat und dann selbst bei laufender Fahrt abgesprungen ist. Ich denke, ich habe die Karre einigermaßen heil nach Hause gebracht, und neuen TÜV hat sie auch noch. Ich habe in unserer recht kurzen gemeinsamen Zeit viel gelernt über physikalische Intuition und einen anderen Blick auf die Welt der Wissenschaft.

Ich danke auch Prof. Andreas Draguhn und den Mitarbeitern seines Labors für die Hilfsbereitschaft bei der Vermittlung von umfangreichem Hintergrundwissen im Bereich Neurophysiologie im Rahmen der Kooperation im Bernstein Zentrum. Ganz besonderes danke ich Martin Both für viele hilfreiche Gespräche, sowohl inner- als auch außerhalb des Whirlpools.

Im weiteren Sinne gilt mein Dank auch dem Fußballteam des IWR sowie der Capella und Camerata Carolina für den bitter nötigen Ausgleich. Insbeson-

dere meine ich damit Gerald und Martina, mit denen ich Freud und Leid der akademischen Welt in vielen Bergbahnsitzungen teilen konnte. Schöne polierte!

Für das fachkundige Korrekturlesen danke ich diesen beiden ebenso wie Tatjana. Ich bin zwar etwas enttäuscht, dass meine bahnbrechende Entdeckung des positiven Chlor-Ions sowie viele meiner geliebten Bandwurmsätze vor euch keinen Bestand hatten, aber die zahlreichen fachlichen und inhaltlichen Hinweise haben doch letztendlich sehr zum Gelingen beigetragen.

Für die regelmäßige Neujustierung meiner neuronalen Konfiguration bin ich Christoph, Philipp und Simon zu nachhaltigem Dank verpflichtet. Unsere diversen Trips nach Ferropolis, Natterthal oder zu Herrn Weber auf Balkonien haben mir geholfen, manch schwere Phase zu überwinden. Wer solche Freunde hat, braucht keine Feinde.

Ich danke meiner Familie, ganz besonders meiner Mutter¹, für die bedingungslose Unterstützung. Die Urlaube im Heizcamp am Ende eines jeden Jahres waren Zäsuren, die es mir ermöglichten einmal abzuschalten und das vergangene Jahr mit einem gewissen Abstand zu betrachten, um dann das kommende mit neuer Kraft anzugehen.

Schließlich danke ich Pia, für einfach alles. Du hast alle Aufs und Abs, scharfen Kurven und die immer länger werdende Zielgerade mitgemacht und mir Halt gegeben. Ich kann mir gar nicht vorstellen, wie es die vergangenen Jahre ohne dich gewesen wäre, und das möchte ich auch nicht mehr.

¹i.e. diejenige welche, die mir das Leben schenkte