Use of Reduction Methods in Multiscale Modeling of Complex Systems

Ruxandra Bărbulescu

A Dissertation
presented to the Faculty
of Electrical Engineering
of University Politehnica of Bucharest
for the degree
of Doctor of Philosophy

Doctoral Committee

Daniel IOAN	Adviser, Prof. at Department of Electrotechnics, University Politehnica of Bucharest
Sorin Dan GRIGORESCU	President, Prof. at Department of Measurement, Electrical Devices and Static Converters, University Politehnica of Bucharest
Gabriela CIUPRINA	Prof. at Department of Electrotechnics, University Politehnica of Bucharest
Wil SCHILDERS	Prof. at Department of Mathematics and Computer Science, TU Eindhoven
Bogdan AMUZESCU	Assoc. Prof. at Department of Biophysics and Physiology, University of Bucharest

December 2018

To Alexia.

Acknowledgments

I would like to thank Prof. *Daniel Ioan* who has been more than a scientific adviser, for his sustained support, enthusiasm and patient guidance in the development of this dissertation, especially for his bold approaches and for teaching me valuable know-how on how to do research.

Also, I am grateful to Prof. *Gabriela Ciuprina*, in whom I have gradually found over the years a teacher, a colleague, a friend; for her strong work ethic and scientific rigour.

I would also like to thank: my colleague *Sorin Lup*, for his unconditional support and eternal optimism; *Mihai Popescu*, for all the discussions on practical applications, Prof. *Valentin Ioniţă* and Conf. *Anton Duca*, for their support as members of my scientific advisory committee all these years.

Andrei Nuţu-Manea, for standing by me with generosity and open-mindedness when the struggle was toughest.

Last but not least, I am grateful to my large family for their unconditional love and support.

The work has been funded by two grants and two research projects as follows:

- the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5 /S/132395;
- the Excellence Research Grants Program, UPB GEX 2017. Ctr. No. 05/25.09.2017;
- the Romanian PN-II-PT-PCCA-2011-3 joint applied research project "Advanced Tools and Methodologies for the Multiphysics Modelling and Simulation of RF MEMS Switches" (ToMeMS, mems.lmn.pub.ro);
- the TD COST Action TD1307 European Model Reduction Network (EU-MORNET, http://www.eu-mor.net/), coordinated by Prof. *Wil Schilders*, within which I visited as a Short Term Scientific Mission the ALGOS group at INESC I&D Lisboa, under the supervision of Prof. *Luís Miguel Silveira*.

The research benefited from the logistics support, computing facilities and research workgroup of the Numerical Modelling Laboratory – LMN (http://www.lmn.pub.ro/), Electrical Engineering Department, University Politehnica of Bucharest, Romania.

Table of Contents

List of Figures

List of Tables

List of Notations

- $\mathbf{D}$ – the electric flux density [C/m^2]
- $\mathbf{E}$ – the electric field strength [V/m]
- $\mathbf{B}$ – the magnetic flux density [T]
- $\mathbf{H}$ – the magnetic field strength [A/m]
- $\mathbf{J}$ – the current density [A/m^2]
- ρ – the electric charge density [C/m^3] / the resistivity [Ωm]
- ε – the dielectric permeability / permittivity [F/m]
- μ – the magnetic permeability / permittivity [H/m]
- σ – the electric conductivity [S/m]
- $\mathbf{P}_p$ – the permanent polarization [C/m^2]
- $\mathbf{M}_p$ – the permanent magnetization [A/m]
- $\mathbf{E}_i$ – the impressed electric field [V/m]
- p – the power density [W/m^3]
- δ – the mass flux density [kg/(m^{2}s)]
- k – electrochemical coefficient [kg/(As)]

Introduction

The main objective of this PhD thesis is to explore the use of reduction methods in multiscale modeling of complex systems to optimally model the phenomena underlying the transmission of information in a neuron and inside the brain. The thesis is focused on the electrical engineering aspects within the Computational Neuroscience, an emerging area of the science and biotechnology. The electromagnetic phenomena are modeled at several scales: ion channels from the Ranvier nodes, myelinated compartments, saltatory conduction along axons. The global models of neurons, synapses, neuron collectivities and the brain modeling are not subjects of this thesis. However, the induced voltage in the brain by Transcranial Magnetic Stimulation systems is also modeled.

The research methodology used here is based on the Bio-Multi-Physics Modeling, using the Computational Science and Engineering (CSE) approaches, in particular High Performance Scientific Computing (HPSC) techniques.

CSE (Fig. 1) is a relatively new multidisciplinary area that deals with the development of computer–based models of natural phenomena and engineered systems. The CSE pipeline encompasses "domain expertise, mathematical modeling, numerical analysis, algorithm development, software implementation, program execution, analysis, validation and visualization of results" [1]. CSE has been described as the "third mode of discovery" (next to theory and experimentation), performing computational experiments to answer questions that neither theory nor experiment alone is equipped to answer [2].

"Historically, simulation has been used as a qualitative guide for design and control, but has often not been expected to provide accurate results for realistic physical systems" [1]. As an emerging discipline, CSE has as goal the achievement of truly predictive scientific capabilities, including but not limited to simulation-based weather and climate prediction, simulation-based design of vehicles and aircraft, simulation-based decisions in computational medicine. It is expected that the problem-solving methodologies and robust tools developed under the CSE umbrella to play an important role in emerging areas such as genomic sciences, computational neuroscience and bioengineering.

CSE differs from mathematics or computer science in that analysis is directed specifically at the solution of problem classes from science and engineering, "and will generally require a detailed knowledge or substantial collaboration from those disciplines. The computing and mathematical techniques used by CSE may be more domain specific, and the computer science and mathematics skills needed will be broader" [1].

The thesis is structured in 7 chapters, excepting this introductory section; the first covers theoretical concepts and the last presents the conclusions. The content chapters

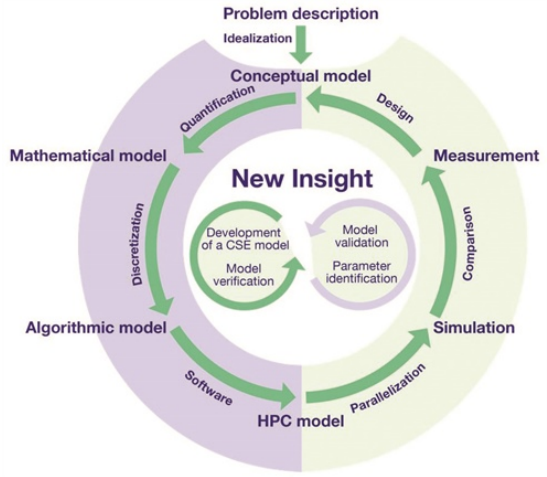


Figure 1: The CSE pipeline, from physical problem to model and algorithms to efficient implementation in simulation software, with verification and validation driven by data. The pipeline is actually a loop that requires multiple feedbacks [2].

$2 \div 6$ are described in the following paragraphs.

Chapter 2 – *State of the Art in Multiscale Modeling of Neuronal Systems* – provides a detailed description of the state-of-the-art in neuronal systems modeling. A multitude of methods have been developed in order to address the complexity of the phenomena taking place in the brain, and multiscale approaches have become an implicit necessity.

One of the simplest models able to describe the electrical signal transmission in a neuronal axon is studied in detail in Chapter 3 – *One-dimensional Models for Neuronal Signals' Transmission*. The model is one-dimensional and the equations are derived from the transmission lines. The analysis goes through the important steps in modeling: conceptual, physical and mathematical modeling, the analytical solution, the numerical modeling, verification and validation of the model. Even if the model is quite simple, the study is fundamental in understanding the underlying phenomena of the signal transmission that will provide the basis for more complex models.

Chapter 4 – *Reduced Order Models of Myelinated Axonal Compartments* – deals with different models of the myelinated compartments (2.5D, 1D) and the reduction of the model's order. The 2.5D field problem is solved using different techniques (analytical, numerical – FEM, FIT, BEM). The most advanced methods of order reduction for the generated models are explored and compared based on the relative errors of the reduced models extracted and their computational efficiency.

In Chapter 5 – *Saltatory Conduction in Neurons* – the phenomena called "saltatory conduction" encountered in the transmission of neural signals along neurons' myelinated axons is modeled and simulated. The models are generated by interconnecting (reduced) models of myelinated compartments with nonlinear models of the Ranvier nodes.

Chapter 6 – *Transcranial Magnetic Stimulation* – studies a medical procedure used to investigate and to treat several brain diseases. An external system of coils with variable currents induces electric field in the brain. Different methodologies to generate realistic geometries are explored and a pipeline to extract them from MRI data is proposed.

Chapter 1

Theoretical Concepts – Multiphysics Modeling

This chapter provides the theoretical background of the multiphysics modeling and correct formulation of multiphysics problems. The problems solved in this PhD thesis follow the modeling procedure described below, with a focus on model order reduction as a central part of modeling multiscale systems. Most of the theoretical concepts introduced in this section are detailed in the next chapters.

The term *multiphysics* is increasingly used to denote a problem that presents one or more of the following attributes (Fig. 1.1):

- "multi-field", meaning the simultaneous excitation and response of the system by multiple physical fields;
- "multi-domain", involving the interaction among continuum representations of systems with drastically different properties (e.g. Fluid-structure interaction, moving solidification boundary problems) through shared boundaries;
- "multi-scale", denoting the consistent bridging of various behavioral models of the system, at various length scales and time rates.

In addition, any combination of these three possibilities generates four more meanings of the term *multiphysics* including the one that reflects the co-existence of all three of them [3].

Any discrete volume in the conceptual attribute space shown in Fig. 1.1 is defined by a triplet of coordinates originating from each one of these attribute axes, and represents a region encompassing certain classes of physical problems. This signifies that these problems can be modeled in a multiphysics sense as defined by their corresponding coordinates [3].

The complex systems that represent most of the open problems nowadays are intrinsically multiphysics. Because the physical processes are interdependent, the key term to describe the new multiphysics paradigm is *the coupling* between several problems, which up to now were approached independently [4].

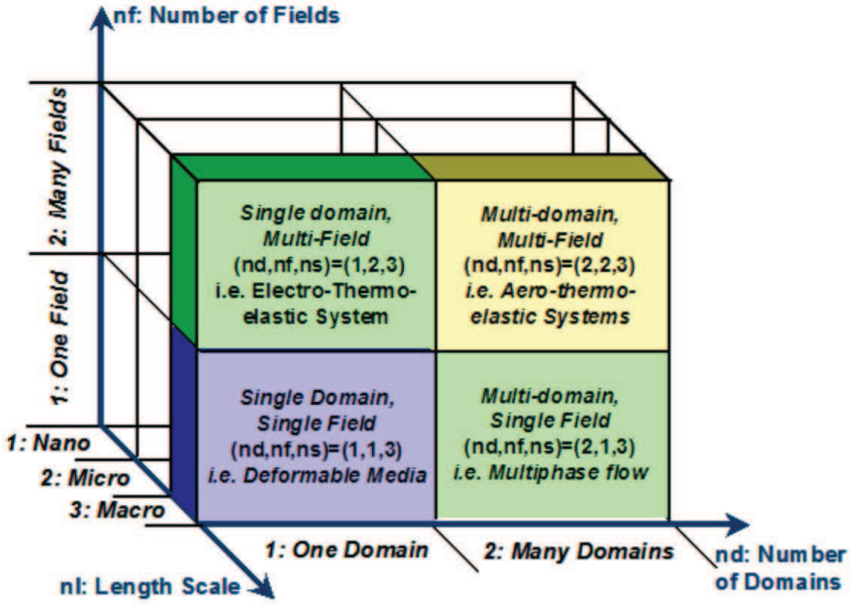


Figure 1.1: Multiphysics attribute space [3].

The modeling and simulation of the systems studied in this thesis take into consideration several coupled phenomena, the following electromagnetic field models being considered, corresponding to different domains:

- Electro-conductive (EC) field in the axoplasm (axon's cytoplasm), where the capacitive effects are negligible;
- Electro-quasi-static (EQS) field through the cell's membrane, where both capacitive (displacement current) and transverse conductive effects (conduction current due to losses) are important.

The saltatory conduction simulation assumes the coupling of a number of similar models, forming a large model with a different length scale than the incorporated models.

The multiphysics modeling procedure described in what follows was developed in the Laboratory of Numerical Methods (LMN) – where most of this thesis research was developed – and comprises the main steps presented below (in this order) [5].

1.1 Conceptual Modeling

This first stage of modeling has the purpose of establishing the geometrical model and the (multi) physical model. This implies describing the structure of the object and then analyzing its grounding principle of functioning.

The ***geometrical modeling*** establishes the size, the shape and the materials of the object and its constituent parts. The coordinate system is set at this point, with a "computational geometrical" perspective in mind, already imagining the data structures, as the model will be eventually described for a computer.

Another goal of this step is setting the computational domain. As many modeling methods impose that their domain of definition to be bounded, the computational domain

will have a border, real or fictive. This truncation is also a simplifying assumption that has to be made explicitly.

The geometry description of every component will obviously involve surface smoothing and/or neglecting some geometric details, which means the idealization of the real forms. All these simplifying assumptions should be recorded in a list at the end of the geometrical modeling, because they are likely to generate modeling errors.

One of the most significant idealizations occurs when assuming that the local quantities (data, unknowns) are not dependent on one or more spatial coordinates. This leads to the following categories of models [6]:

- **1D**: the local quantities depend on only one spatial coordinate (usually Cartesian);
- **2D**: the local quantities depend on two spatial coordinates; if these coordinates are Cartesian, the model is called plane-parallel;
- **3D**: the quantities depend on three spatial coordinates;
- **1.5D**: the local quantities depend only on r – the radius of a cylindrical coordinate system; mathematically the model is 1D but from the physical point of view we talk about a 2D model; this model is both plane-parallel (along Oz axis) and axisymmetric (Oz is the axis of symmetry);
- **2.5D**: the local quantities depend on the spatial coordinates z and r of a cylindrical system; this means the model is still axisymmetric (with Oz the axis of symmetry); the model is 2D mathematically and 3D physically;
- **1D/3D**: the local quantities depend only on the coordinate r of a spherical system (models with spherical symmetry); mathematically the problem is 1D whereas physically it is 3D.

The ***physical modeling*** means identifying the fundamental physical phenomena the functioning is based on. The main physical quantities are identified, as well as the causal relations between them, describing the state and interactions within the model. The physical phenomena lead to physical laws that lead to equations, which from the spatial point of view have to be in accordance with the dimensional category established during the geometrical modeling.

From the time point of view, the type of variation determines the operating mode (also the physical regime in which the field functions) and the subsequent type of simulation. The regimes usually considered are: **stationary** (assumes that the quantities do not depend on time); **harmonic** (the quantities have a sinusoidal variation, with the same frequency f); **periodic** (the variation is recurrent, with the same period T); **transient** (the variation exists, but it is not known); **modal analysis** (seeks the eigenmodes of oscillation, which can occur in some structures after the excitations end). For the electromagnetic field there is another differentiation, between the **static** and stationary regimes, the first being a stationary regime without energy transfer. The **quasi-stationary** regime is also used (inductive and capacitive); in this regime the quantities vary so slowly that some effects can be neglected (such as the electromagnetic induction phenomenon or the displacement currents effect).

The problem's solution can also be influenced by other characteristic data, such as material constants and field sources, in which case this data appears in the equations and has to be explicitly assessed. If the field sources are internal, then they are usually described by the local physical quantities of the corresponding computing domain. The external field sources are described using the boundary conditions, whereas the anterior sources are described by the initial conditions.

In some cases the physical fields have different operating modes in distinct subdomains of the computing domain. These should be identified, as they may be a valuable resource for the subsequent order reduction of the extracted model. For multi-domain problems it is also important to determine the relations describing the coupling between domains (transfer relations, inter-connections). Together with the identification of the field regimes and the field sources, establishing the coupling relations represents an important step for making the transition from a qualitative description of the model to a quantitative one, which is the main goal of the physical modeling stage.

1.2 Mathematical Modeling

This stage of modeling aims to formulate the problem in mathematical terms and verify that the problem is well formulated. For a direct problem, its initial form means solving a system of PDEs. Conversely, the verification of the well formulation conditions enforces the reformulation of the original problem, which usually means passing from the strong formulation to the weak form of the differential equations (also called variational form). The reformulation has additional advantages as it facilitates the proof of some theorems and the development of numerical approaches. In the weak form, the solution is no longer a classical function, but a generalized one, for which the value in a point of the computing domain has no meaning. This profound transformation requires the use of modern mathematical approaches, such as functional analysis applied to PDEs and therefore to the spaces of functions such as Lebesgue square-integrable, with generalized derivatives or Sobolev [5].

The problem's initial formulation requires the identification of the given data, the identification of the unknowns and the equations that connect them. The given data is comprised of:

- Information related to the spatial computing domain (shape and size), in mathematical terms;
- Data describing the materials in every point of the computing domain; if the materials are linear this data is represented by material constants (real numbers for isotropic and tensors for anisotropic materials); for nonlinear materials the characteristic functions are given;
- Internal field sources in every point of the computing domain, which are sometimes included in material characteristics;
- Boundary conditions, describing the effect of external field sources; they should be known in every point on the border and at every moment in time (in the interval of

simulation);

– Initial conditions, describing the effects of anterior evolution of the system; they are required only for transient analysis.

The unknowns in a field analysis problem are local physical quantities, mathematically represented by functions defined on the computing domain (spatial and temporal).

For a field analysis problem the well formulation means the concurrent fulfilling of the following three conditions [7]:

– The existence of the solution – the problem must have a solution for every excitation from a class of functions;

– The uniqueness of the solution – the problem must have a single solution in the spatial and temporal computing domain;

– Well-conditioning of the problem – the solution depends continuously on the problem given data, small deviations of data do not generate excessive deviations in the solution (the error is not excessively amplified).

The first step in proving that a problem is well formulated is the precise identification of the space containing the given data and the space where the solution lies. This action is called "setting the functional framework" of the problem to be solved.

The weak formulation is obtained by Galerkin projection on a test functions space, which is isomorph with the trial / base functions space (the space where the solution is searched for). In the static cases, the identification of the trial functions implies making the distinction between natural boundary conditions (described in the expression of the weak form of the equations – Neumann type) and essential boundary conditions (imposed on the solution from the start, therefore satisfied by the trail functions – Dirichlet type) [5], [6].

1.3 Analytical modeling

Most complex problems cannot be solved analytically. However, it is recommended that before the numerical solving an analytical solution to be found, even if this means the oversimplification of the original problem. The approximate analytical solution is used to validate numerical methods and to better understand the way the solution depends on the input data, by sensitivity analysis. Examples of analytical modeling are in what follows [5]:

– Simplifying the geometry, usually to a 1D problem;

– For transient problems, finding the stationary solutions, which describe the initial and final state of the system;

– For nonlinear systems, their linearization and subsequent application of perturbation analysis, by considering small variations around the stationary solution [8];

– For harmonic and transient problems, the analysis in the frequency domain (complex/operational representation);

– Applying finite differences method for the spatial derivatives; the system obtained is discrete, with a relatively small number of degrees of freedom and an approximate analytical solution; this is a *lumped parameters* system, whereas the original system has *distributed parameters*;

– Applying separation of variables [9]; in order to find an analytical solution the equation is rewritten so that each of the two variables occurs on a different side of the equation.

The approximate modeling requires additional simplifying assumptions that are unlikely to be realistically satisfied. However, the outcome of this stage of modeling is getting a sense of the interactions within the model and its behavior in particular conditions.

1.4 Numerical Modeling

Numerical approaches are mandatory to find solutions to complex problems. The most important numerical techniques are the following:

– **FEM** – Finite Element Method [10], [11], [12];

– **FDM** – Finite Differences Method, or some of its versions, such as Finite Volumes Method (FVM) [13], [14] and Finite Integration Technique (FIT) – the corresponding approach in electromagnetism [15], [16], [17];

– **BEM** – Boundary Element Method, also known as the Method of Moments (MoM) [18], [19].

Table 1.1 synthesizes the characteristics of the methods enumerated above.

Table 1.1: The numerical methods compared (adapted from [5]).

Method	Discretization mesh	Equation form	System matrix
FEM	Unstructured, composed of triangles, quadrangles, tetrahedra, etc.	Weak, differential form	Sparse, symmetric, positively defined, diagonally dominant
FDM/FVM, FIT	Grid with regular topology (tensor product of 1D meshes); in the case of hyperbolic equations, pair of interlaced dual networks.	Differential / global form	Band matrix, sparse, symmetric, diagonally dominant
BEM	Unstructured 2D meshes on domain border or on interfaces between homogenous subdomains.	Integral form	Full, positively defined

These numerical methods are extensively used to solve complex problems and for this reason many software packages have been developed, both general purpose and specialized on a particular set of problems. To select the most suitable modeling methodology for a certain problem it is necessary to understand thoroughly the characteristics of the analyzed model and to be acquainted with the existing software procedures and tools.

1.5 Extraction of Reduced Models

A comprehensive discussion about the necessity of model reduction in modeling multiscale systems is done in Chapter 2.

From the simulation time point of view, the complexity of a model is quantified in the number of degrees of freedom (DoFs), which is in close connection with the number of inner state variables of the model and the complexity of the discretization mesh. In the general sense, model reduction aims at finding a model that retains to some degree the behaviour and the predictive power of the original model, but has much lesser DoFs and is therefore much easier to simulate. Concurrent advances in computing power and simulation algorithms may appear to make model reduction a less essential process than it was in the past [20], but in reality model reduction is as important as ever with the growing ambition to simulate higher complexity models, especially in biology, where the level of detail can generate models with thousands of state variables [21].

We can dissociate two phases in the process of model reduction. The first is based on **discretization**, which transforms an infinite model to one with a finite order. Discretization can be performed at several levels. For example, the solution of a linear, parabolic PDE determined by separation of variables is a series of functions with exponential variation over time, with increasingly smaller time constants. We basically evaluate the solution by truncating the series to a finite number of terms. These are the solutions of linear ODEs, therefore by switching from PDE to ODE we perform a first reduction of the model [5]. Another form of model reduction is expressed through numerical methods such as FEM, as the inherent discretization transforms the original model to one with a finite number of states. Moreover, all computational models require discretization not just for the state variables but also for the boundary conditions, initial conditions, input and output data (discrete signals). We can state that passing from a continuous mathematical function to a data structure such as a vector or a matrix (discretization) denotes and implies model reduction.

The second phase refers to **model order reduction** (MOR) of the mathematical system of equations. The reduction of the original model order aims at identifying a system described by ODEs with a number of states $q \ll n$ but having the same number of inputs and outputs as the original system and providing a relationship between the input and output signals close to that of the original system. Also the existence of an error bound is expected. In some situations it is desirable to preserve other properties of the original system such as passivity or stability. The reduction procedure should be computationally stable and efficient [22].

Many mathematicians have contributed to developing new and/or improved MOR

methods, as this subject has generated a tumultuous and fertile literature in the last decades [23], [24], [25]. There are two sets of methods that are currently in use, namely: SVD based methods and moment matching based methods. One commonly used approach is the so-called Balanced Model Reduction [26], whereas for nonlinear systems Proper Orthogonal Decomposition (POD) is more suitable, both representing the former category. From moment matching based methods Krylov subspace methods are to be mentioned. All the methods analyzed and applied in this thesis are formally described in Chapter 4 and Chapter 5.

1.6 Models' Verification and Validation

This is the final step of the modeling activity and it completes the circle of modeling by returning to reality. Model verification checks if the solution satisfies the equations and model validation ensures that the model formulation is in agreement with the reality, in other words that the equations are correct.

An important issue in **model verification** is the formulation of deviation indicators, in other words what are the criteria of interest and for each criterion how do we define the deviation. In order to verify the solution of a field problem, the problem is solved for a particular excitation (input, field source) and it is determined to what extend this satisfies the field equations and the boundary conditions. Most often very specific or even degenerated shapes are used for the excitation signal, such as step/pulse functions or even constants. For these particular cases we are able to determine the analytical solution (see the third step of the modeling procedure) that will act as the reference. During the model reduction stage, the reduced model is compared with the original model, from which the quantities of reference are chosen.

The result of the quantitative evaluation will be a vector, however it is desirable to have only one numerical criterion to synthetically express this difference, i.e. to measure the distance between what should be and what is actually obtained. From a mathematical point of view, this dilemma is reduced to choosing a vector norm. Most often the error is relative, as ratio between the deviation norm and the norm of the vector characterizing the quantity of reference. The most common vector norms used in practice are the Euclidean norm (square root of the sum of the squares of the components) and Cebisev norm (the maximum absolute value of the components). The first describes the mean square deviation and the second describes the maximum deviation. The Euclidean norm takes into account all the components, whereas the Cebisev norm refers only to the most aberrant component. In order for them to be comparable the Euclidean norm should be computed as an average (thus dividing the sum of the components' squares by their number before computing the square root). In some cases some components are more significant than others, therefore a weighted norm should be defined.

In conclusion, the error definition plays a big role in the interpretation of the results and should be under discussion for every particular case. For example, for a system of equations the relative error can be defined as the ratio between the residue (the difference between the free term and the vector obtained by replacing the solution in equations) norm

and the free term norm. This error however might not be relevant for the quantitative evaluation of the solution's correctness, since there are many examples with a reasonably small residue, but a significant actual deviation of the solution. This typically happens for ill-conditioned systems, where small differences in the given data (field sources, described by the free term) may cause big differences in the solution. The condition number should be determined in order to estimate the solution's variability, meaning the rate at which the solution will change with respect to a change in the data. Depending of the equations, the procedure may involve the analysis of the system matrix (conditioning is a property of the matrix), sensitivity analysis (may be empiric), parametric analysis.

The last and possibly most important phase in modeling is the **model validation**, which means confrontation with the reality. The validation can be performed experimentally, but prototype design and execution may be very expensive and in some areas where prototyping is not an option, such as biology, ethical issues may arise. The model can be validated by the comparison with other models in the literature, preferably already validated by experiments. The tests should consider the behavior in typical conditions and the behavior in particular situations. For the particular situations the analytic-approximate model may be used as reference. The model should be able to reproduce at least qualitatively the behaviour in the particular situations considered in the analytical approximate model. For complex problems however the results may be significantly different, but the differences should have plausible explanations.

Chapter 2

State of the Art in Multiscale Modeling of Neuronal Systems

This chapter is dedicated to the state of the art in modeling of biological systems in general and neuronal systems in particular. Model reduction is emphasized as an essential part of efficient modeling, in the presence of the intrinsic particularity of *multiscale* of complex biological systems [27].

2.1 Multiscale and reduced modeling of Biological Systems

Until recently, biology was a subject only for biologists. But as a change of paradigm had taken place worldwide and many research fields had become interdisciplinary, the research domains related to biology have aligned to this trend. There is a wide variety of emerging fields involved, such as molecular biology, electro-biology, computational biology, evolutionary biology, systems biology. All these are strongly related with the "omics" sciences – genomics, proteomics, interactomics – having as result an "olome" – genome, proteome, interactome – which aim at the collective characterization and quantification of pools of biological molecules that translate into the structure, function, and dynamics of organisms [27].

By this process, systems biology aims to be a system-level understanding of biology. In the first book on this subject [28], Hiroaki Kitano said that systems biology "is intended to be biology for system-level studies, not physics, systems science, or informatics, which try to apply certain dogmatic principles to biology". Systems biology is a holistic approach to biology, a bold attempt to understand not only the structures, but also the dynamics, control and design methods of complex biological systems. The main reason is that the components of biological systems act differently in isolation than they do when integrated into a larger system [29]. As every interdisciplinary field, systems biology makes use and becomes a pretext for the advances in theory, experiments and computational modeling. Therefore systems biology became a hybrid; its theoretical foundation consists of quantities, principles, laws and techniques from other fields such as biology, physics,

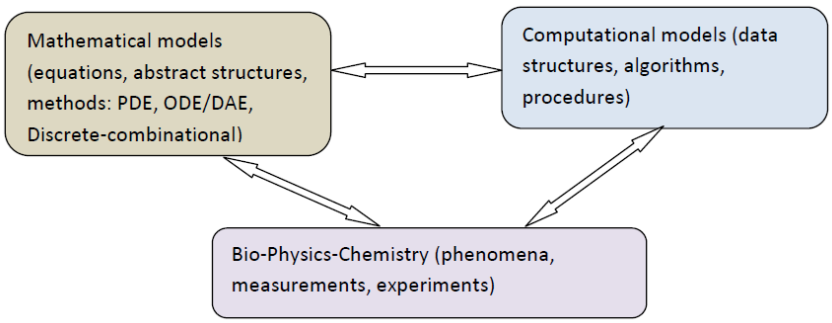


Figure 2.1: The mapping between the three representations of biological systems.

biochemistry, mathematics, control and computational sciences. But their consequences are beyond those of each discipline involved.

The interdisciplinary character of systems biology is underlined by the interactions of its representations in the three worlds: the reality, the abstract and the virtual representations. Any complete model should have these three related representations, in total accordance with CSE. They are in strong relationships (Fig. 2.1) with mutual influences.

A suitable mathematical model should not just fit the experimental results, but should explain the underlying phenomena, make predictions and generate experimentally testable assumptions. These kinds of quantitative models are the theoretical fundament of valuable computational models. Advances in IT allow the implementation and development of more sophisticated models. However, the explosive evolution of computers cannot substitute advanced abstract – mathematical modeling activity and deeper understanding of the reality [27].

The biological complexity relies first in the fact that the collective properties of a system are not equal to the sum of individual properties of its contained sub-systems. This is the main reason for trying a system-level understanding of biological systems. Moreover, complex systems are not static and their present behaviors are in part determined by their history [30]. This is why one cannot eliminate time from the complete analysis of biological systems.

Until the year 2000 most of the system-level analysis attempts were made at cell level [31, 32, 33, 34]. The idea behind a system understanding at cellular level was to not only identify the components (ex. genes, signal transduction networks, physical structures), but also to consider the cell as a system, to try to determine the interaction between these components and to describe the system properties, such as its dynamics, control and design methods.

In the literature there are many attempts to analyze the modeling process in systems biology, many of them stressing to the importance of multiscale approach [35, 36, 37, 38, 39, 40, 41, 42, 43, 44]. However the model reduction is not always properly identified as an essential procedure for an effective multiscale analysis.

The multiscale approach to biological systems is based on the idea that the properties of a system at a certain scale (e.g. cellular scale) are influenced by the interactions at lower and higher scales (molecular and tissue-organ scales). In order to predict quantities

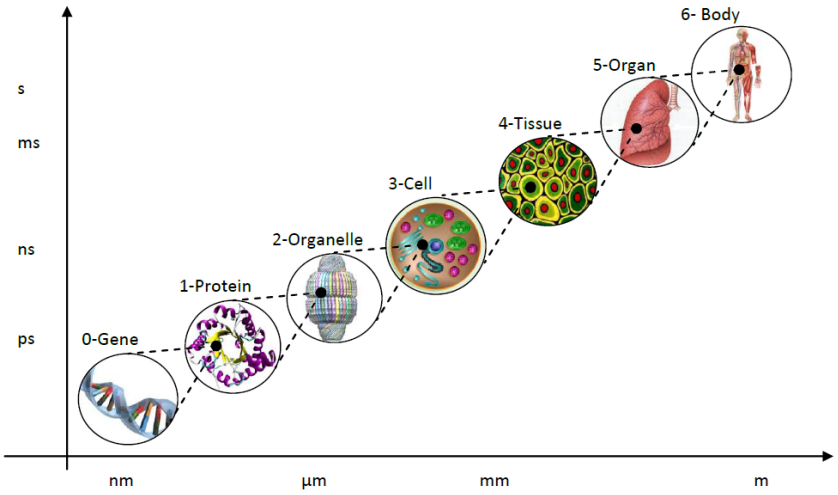


Figure 2.2: Hierarchical levels in multiscale biological modeling.

at a macro level, it is necessary to use the information at a lower-scale. A system-level understanding of biological systems (systems biology) implies a multiscale approach, for "this is the uniform nature to incorporate complex bio mechanisms" [30]. Reference [45] even defines systems biology as quantitative, post-genomic, post-proteomic, dynamic, multiscale physiology.

The idea of multiple scales in biology is strongly connected with the principle of biological organization on a hierarchy of scales from atomic level to organism, population and ecosystem. Each level in the hierarchy brings an increase in complexity that is described using the concept of emergence. Each level presents emergent properties – coherent structures, patterns and functions – which are not present or are irrelevant at lower levels. Emergent properties reside in the properties of the ensemble rather than of any individual state. The challenge is to determine the interactions between scales and to conserve relevant information from lower scales to higher scales [27].

A biological system should therefore be perceived as a four-dimensional entity spanning over many spatial and time scales. In the three-dimensional space we can clearly differentiate the molecular ($10^{-10} \div 10^{-7}$ m), the cellular ($10^{-6} \div 10^{-4}$ m) and the tissue/organ level ($10^{-2} \div 1$ m). In the time domain biological systems range from the molecular interaction timescales ($10^{-9} \div 10^{-6}$ s), through the cellular (10^{-1} s) to the macro ($10^{5} \div 10^{9}$ s: days to years) scale [38, 39, 42, 46]. The spatial scaling factor – of 9 orders of magnitude – considered in Fig. 2.2 is unidirectional; but in the 3D physical space it becomes triple. Consequently, the number of elementary structures from the first level contained in an organism is 10^{27}, which makes it impossible to model a biological system from the last level by exclusively using structures from the molecular level. This is the obvious argument for the absolute need of a multiscale approach, even for just two consecutive levels, having a scaling factor of $10^{4} \div 10^{6}$. The obvious conclusion of this image is the tight link and correlation between multiscale (several spatial degrees of refinement) and multi-rate (different speed of events, in time). Each spatial level has its characteristic time constant.

Although at the higher scales there can be uncovered some universal characteristics to all biological systems, the lowest levels (genome, transcriptome, proteome, metabolome)

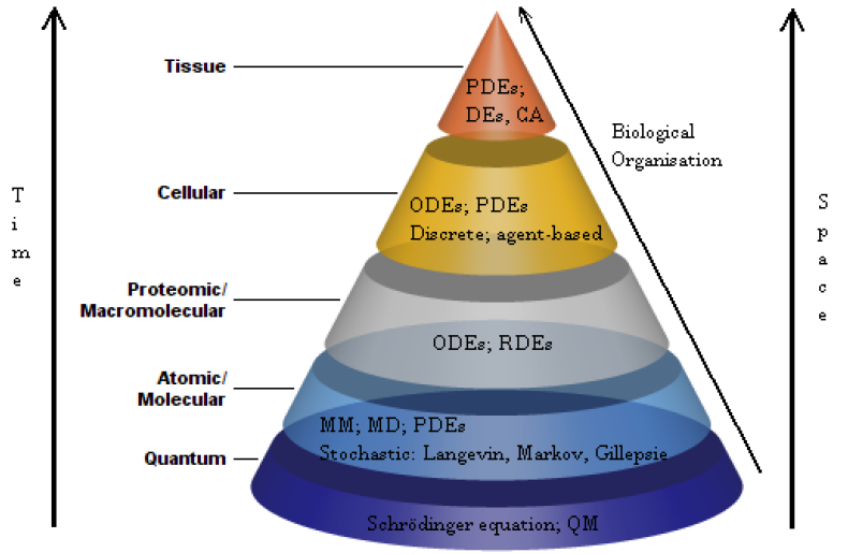


Figure 2.3: Abstraction levels of hierarchical biological organization.

exhibit organism specificity [47]. Every spatial scale has its specific mechanisms, happening at their own time scale, so it has to be modeled with a suitable abstraction level (Fig. 2.3).

Different scales imply different physical models, which imply different mathematical models which imply different quantities. What happens at the genome level is described by the quantum mechanics; at the cell metabolism level the biochemistry plays an essential role while at the membrane level the electro-chemistry is important. There has to be a multiphysics approach, and the key issue is to find the appropriate representation of the behaviors at different scales and the proper interface between scales. It is apparent that a model based on a single type of physics and using a uniform spatial scale would not be capable of describing this multitude of biological processes and providing fundamental understanding [44].

The diversity of methodologies and techniques applied in mathematical modeling and the computational representation and simulation for different space and time scales of biological systems are presented in the Table 2.1. These may be:

- **Differential ODE/DAE** equations in which time is the independent variable, able to describe the dynamical behaviour of several variables. Generally they are nonlinear, implicit or explicit, but many times they are linearized, mainly for an easier solvability:

$$\text{DAE}: f(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0 \Rightarrow \mathbf{E}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \text{where } \dot{\mathbf{x}} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}, \mathbf{x}(t) \in \mathbb{R}^n \text{ are state variables;} \quad (2.1)$$

$$\text{ODE}: \dot{\mathbf{x}} = g(\mathbf{x}, \mathbf{u}) = 0 \Rightarrow \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \text{where } \mathbf{u}(t) \in \mathbb{R}^p \text{ describes stimuli.} \quad (2.2)$$

  They are used to model the molecular dynamics of proteins, the dynamics of biochemical reactions and many other biological aspects. By using these equations – at

the cellular level – biological regulation networks can be modeled as electrical circuits where signals are produced, propagated and sensed. Spice-like circuit simulators such as Xyce may be used to simulate large control networks consisting of entire cells or cell cultures in order to understand the dynamics and stability of such systems [48].

- If there the spatial reaction-diffusion phenomena or heat and mass transfer are modeled, then **PDE equations** are used, usually of parabolic type:

$$\text{PDE} : f(\frac{\partial \mathbf{x}}{\partial t}, \nabla \mathbf{x}, \mathbf{x}, \mathbf{u}) = 0 \Rightarrow$$
$$\Rightarrow \mathbf{D}\frac{\partial \mathbf{x}}{\partial t} + \nabla(\mathbf{C}\nabla \mathbf{x}) + \mathbf{B}\mathbf{u} = 0,$$
$$\text{where } \mathbf{x}(r,t) \in \mathbb{R}^n \text{ are state variables.} \quad (2.3)$$

  Other PDE equations are used, such as those of electric field, Navier-Stokes to model several fluid dynamics or mechanical equations of solid body equations to model structural elastic or viscous-plastic deformations and Schrödinger equations at the atomic level.

- **SM: Stochastic methods.** The thermodynamic fluctuations of the proteins accumulated in biological systems require the use of appropriate stochastic techniques to model the cells and tissues. They are applied in the simulation of dynamical aspects of chemical reactions, as in the well-known SSA (Stochastic Simulation Algorithm, developed by Gillespie in 1977, based on Markov transitions), CME (Chemical Master Equations, developed by Nicolis and Prigogine in 1977) or probability distribution described by Fokker-Plank PDE equation (developed by Risken in 1989).

- **DM: Discrete methods** are more computationally efficient than the continuous methods. They are based on temporal or spatial discretization, drastically reducing the number of variables or the number of bits needed to represent a variable. Difference equations (aka iterated maps) may be used:

$$\text{DE} : f(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}) = 0 \Rightarrow \mathbf{x}_{k+1} = g(\mathbf{x}_k, \mathbf{u}),$$
$$\text{where } \mathbf{x}_k \in \mathbb{R}^n \text{ are discrete state variables.} \quad (2.4)$$

  Cellular automata (which describes the interactions between elements which are distributed over a spatial grid), Boolean networks (with elements having random state and input, described by Boolean functions), agent-based methodology (an extension of cellular automata, but their elements are agents complying more complex rules) are other types of discrete techniques, known as rule-based modeling. Even if the elements follow very simple rules, due to their interactions the entire community has a complex behavior. Compared with the continuous models, they are easier to simulate, but the result is less accurate.

The interdependence between two neighbor scales is a causality arrow in both directions (Fig. 2.4). For example ion channels → whole-cell → heart [42, 52] and in reverse, proteins are modulating gene expression [29, 42, 53]. This interdependence can be viewed as

Table 2.1: Space and Time Scales, Mechanisms and Representative Modeling Methods. Codes: **Deterministic**, *STOCHASTIC*, Discrete, Different Interpretations. Abbreviations Used: RDE: Reaction-Diffusion Equation; ODE: Ordinary Differential Equation; PDE: Partial Differential Equation; MD: Molecular Dynamics; QM: Quantum Mechanics; MM: Molecular Mechanics; ECM: Extra-Cell-Matrix, [36, 37, 42, 49, 50, 51]

	Space scale	Time scale	Mechanism	Modeling Approach/Methods
Quantum	10^{-10}	10^{-15}	electron – electron interaction	Schrödinger equation QM
Atomic / Molecular	10^{-9}	10^{-12}	structural determination of atom-atom force fields	**MM**
			dynamics	**MD: Newton's laws**
			dynamics + diffusion in space	ODEs + RDEs → PDEs
			Brownian motion	*Langevin equation*
			ion channels "gating" – electrical activity	**Hodgkin & Huxley formalism** *Markov chains (transitions) – Gillepsie's algorithm*
Proteomic / Macromolecular	10^{-6}	10^{-9}	gene expression / regulation	ODEs (law of mass action)
			signaling pathways / networks	
			protein – protein interaction networks	
			pattern formation	RDEs
Sub-cellular and Cellular	10^{-4}	10^{-5}	metabolic reaction networks	ODEs (law of mass action)
			action potential	ODEs
			flux of substances inside cell – convection, diffusion	PDEs; spatial variable density functions
			flux of substances through cell membrane	ODEs (conservation laws)
			whole-cell Ca transient in Ca signaling	agent-based
			thermodynamic fluctuations	*Gillepsie's algorithm*
Tissue	10^{-2}	10^{-3}	cell-cell interactions	mechanical models
			cell-ECM interactions	
			cell differentiation (with position within the tissue)	PDEs cellular automata
			electric wave conduction (cardiac, neurons)	PDEs(RDEs)
			blood flow, coronary circulation	Navier-Stokes equations
			cardiac conduction	difference equations (iterated maps)
Organ and System of organs	10^{-1}	10^{-1}	Physiology of systems: skeletal, muscular, cardiovascular, nervous, digestive, respiratory, endocrine, lymphatic/immune, excretory, reproductive.	integrating time models together, by consideration of real geometry of the organ or system of organs

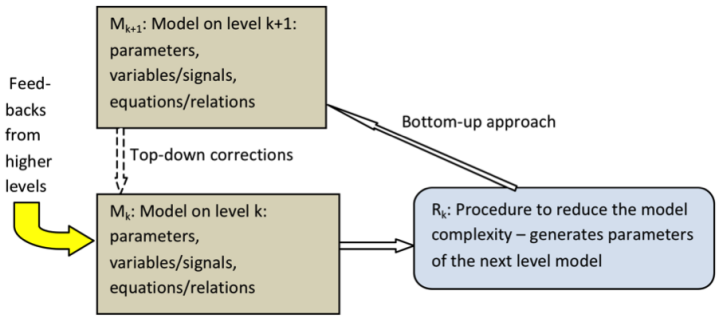


Figure 2.4: Multiscale modeling and model reduction.

bidirectional coupled problems where in addition the two coupled systems are modeled using different physical and mathematical relationships. Interactions can therefore occur at the same scale or between scales, resulting in feedback loops between scales. There is a clear difficulty in connecting these models in a correct and relevant way.

There are different strategies for the integration of several levels. They are sometimes categorized according to the orientation, in Bottom-up and Top-down approaches. If we consider for example only two adjacent scales, the Bottom-up approach tries to determine the system's behavior on the higher scale by analyzing the dynamics and interactions of its components at the lower scale. These kinds of models are adaptive and robust, but they are computationally intensive, especially if we have to consider more than just two scales [42]. The Top-down approaches try to reverse-engineer underlying mechanisms (lower-scale) from higher-scale observations. This approach can lead to a simpler and intuitive model, where hypotheses can gradually increase their level of detail directly backed-up by the data; but this model is less robust and one can argue that higher scale phenomena may have multiple different potential underlying explanations on more fundamental scales, so the interactions can become ambiguous [53].

In many models of physical systems there is an implicit multiscale modeling, where governing equations capture the relevant properties of lower spatial and temporal scales. As at this moment there are no governing equations for biological systems, we need to explicitly consider assumptions across multiple spatial and time scales.

Several modeling challenges implied by a multiscale approach are described in [43]. According to [42], significant conceptual and practical gaps exist between scales, some of them being irreducible, such as:

- Keizer's paradox (an example of a system which modeled with SM always undergoes extinction $x = 0$, while it is continuous-deterministic modeled, it tends to a nonzero steady state);
- Noise-induced phase transitions (inducing qualitatively properties to systems, in certain conditions);
- Crucial differences between 0D and 3D models (e.g. accuracy in modeling the spatiotemporal dynamics of cells).

The authors have identified several techniques to close-up the gaps between scales, such

as:

- **Mean field theory** (low-dimensional representation of high-dimensional systems developed in statistical Physics to compute the macroscopic values of microscopic variables);
- **Coarse graining** (an ensemble of neighboring elements are represented by their average behavior, numerical discretization being a case of coarse graining);
- **Nonlinear dynamics** (allow the description of complex behavior of low-dimensional systems such as chaos, cycles, bifurcation, self-organization);
- **Systemic biology approach** (holistic modeling to allow the identification of feedbacks between several scales, which are characteristic to the living systems);
- **Hierarchical modeling** (reflecting the physical and spatial organization of the organism, from the intracellular to the intercellular level and beyond – tissues, organs, etc.): identifying the dominant sub-system [54] or the use of Petri nets [43].

In the context of coarse-graining, different sub-techniques have been studied, such as the Henderson theorem (which states that under rather weak conditions two pair potentials which give rise to the same correlation functions cannot differ by more than a constant), the Inverse Monte Carlo method [55] or the application of self-organizing maps (i.e. a neural network type approach) for the analysis of conformations of molecules of soft matter [38].

The process of identifying the different spatial/time scales for a complex biological system (encapsulation) might be the most difficult part in the modeling process. Considering the hierarchical organization of biological systems, one can model a whole organ by putting together millions of cells, every cell consisting of thousands of ion channels. But this type of model will be by far numerically intractable using the existing computational power. This is why there is a close connection between multiscale modeling and model reduction. In order to model multiscale systems it is mandatory to use different spatial/time resolutions. One can create a medium high-dimensional scale model using a high resolution model for a low-scale, which will generate many degrees of freedom for the medium-scale model. Then one reduces the complexity of the medium-scale model by creating a low-dimensional model which can be incorporated in a high-scale model that makes use of the relevant information from the lowest modeled scale. Table 2.2 presents the main multiscale methods for bio-systems modeling.

Model reduction techniques for deterministic models are classified in several ways. For example in [54] they are:

- **Trajectory based techniques**: tries to find a small number of reduced variables from integration of the dynamical equations (Proper Orthogonal Decomposition – POD);
- **Singular perturbations techniques**: slow/fast decomposition (or master-slave splitting, because it eliminates the fast variables whose dynamics is slaved by the slower variables;

Table 2.2: Multiscale Methods in Modeling of Biological Systems [27]

QM / MM	the region of interest is represented Quantum Mechanically; surrounding protein/lipid/solvent is represented at atomic scale with an empirical derived potential function.
Coarse-graining	represents an idealization of various kinds; mean-field theory, rule-based modeling and numerical discretization can be considered as coarse-graining methods ; also treating the cell as the smallest unit.
Mean-field	can be considered a particular version of coarse-graining; treats a system by assuming that the elements are controlled by a single mean field, which is created by all the elements themselves [42]; appropriate when the elements are globally coupled or when the system is well mixed.
Non-linear dynamics	iterated maps; manifold reduction; slaving principle.

- **Aggregation or lumping techniques**: replace the reactions mechanism by simpler mechanisms in which some intermediate species are absent.

However [23, 56, 57] present different classifications, such as projection methods and rational interpolations/fitting in the frequency domain. The goal of multiscale approach is not only to develop models at several scales; they have to be linked in such a manner to allow the transfer of information between levels. It is a challenging task. Results obtained at each level have to be analyzed and simplified in order to be useful for the higher level. This actually means the reduction of the model complexity. For instance, if the current model is a continuous one in space and time – described by PDEs – it may be discretized in order to be described by ODEs or DAEs. It is an important reduction, because the system order goes from infinity to a finite number. Another example can be the model order reduction from a high value to a lower one, as it happens in classical MOR of systems described by DAEs, such as Krylov space methods, TBR or Rational interpolation. Moreover, the change of modeling methodology, from complex to a simpler one, e.g. PDEs to DAEs to ODEs to SMs to DMs as well as by linearization are definitely procedures to reduce the model complexity and may be used as bridges between several scales.

The parameters of biological networks can be extracted in a manner similar to the extraction of reduced model from the electromagnetic field model with EMCE boundary conditions [58]. By using Adaptive Frequency Sampling and Vector Fitting (AFS-VF), the parameters of lumped circuits are extracted in parallel [48], equivalent to the components with distributed parameters [59]. A similar approach is presented in [60], applied to the thermal phenomena. Both [58] and [60] reveal the major importance of the terminal reduction for an efficient complexity reduction. In this approach, the PDEs which describe several bio-physical and bio-chemical fluxes are numerically solved by using advanced HPC techniques suitable for multiscale and Domain Decomposition [61], [62], [63]. Multiscale

and model reduction are therefore combined in a very efficient manner and the problem becomes a complex multiphysics one, with all three attributes described in [3]: multiscale, multi-field and multi-domain. The multipolar element with distributed parameters of biological networks is the key concept. Such element is encapsulated in semipermeable membrane or it is bordered by a virtual boundary. It describes the multiscale mass transfer of several species, their flow or diffusion and reaction, as well as the electric potential distribution and the electric current flow. Consequently, on the element boundary two kinds of terminals are defined: electric and flow terminals. They represent exactly the circulatory and neural structures that are the main concern in the transplant surgery. These ideas are developed in [64].

2.2 Multiscale Modeling of Neuronal Systems

Neurons are the basic cells of the neural system, the most complex system in the human organism. They consist in a soma, a dendritic tree and an axon ended with the axonal tree, which are interconnected by means of synapses with the dendrites nodes of other neurons (Fig. 2.5). Myelination is an essential process in the formation of the nervous system. It begins before birth and continues until adolescence, resulting in increased neuronal performance, primarily in the transfer of neural signals along axons. Myelination consists of wrapping axons with an electrically insulating layer, built by a series of glial cells. These cells have a lamellar shape and spiral wrap around the axons (Fig. 2.6), being composed of myelin, a protein-rich fat substance with good insulating properties. The space between two myelinated sections is called a Ranvier node. Having a high density of ion channels, these nodes are essential in regenerating the neuronal signal, previously attenuated during the transmission across myelinated compartments. The electrical phenomena occurring are linear in the myelinated sections and nonlinear in the Ranvier nodes. The neuronal signal transmitted along the myelinated axons seems to jump from node to node, a phenomenon called saltatory conduction. Among the first papers describing the saltatory conduction, [65] also presents how the characteristic parameters of the nerve fiber can be experimentally determined [66].

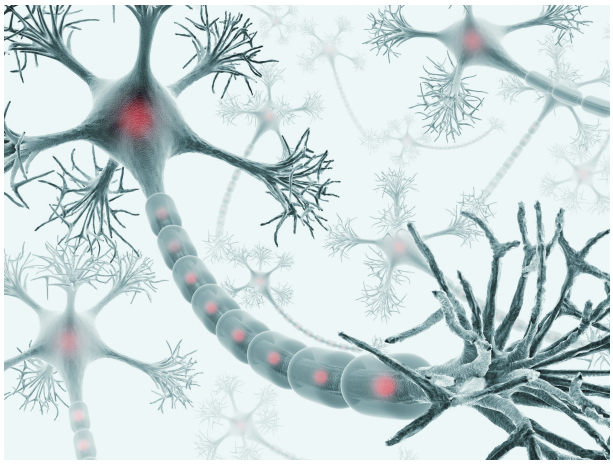


Figure 2.5: The neuron as basic cell of the neural system [67].

Although many consider the delay of inter-nodal transmission negligible to the delay of signal reconditioning in the Ranvier nodes, it has been concluded that things are not exactly so [68]. Extracting small size models with acceptable accuracy is essential for the simulation of saltatory conduction and consequently for the efficient simulation of impulse neural circuits, which are very complex circuits in the central and peripheral nervous system. In order to extract precise models of myelinated compartments, an exact knowledge of their morphology and material characteristics is necessary, and also the complete understanding of the physico-chemical, bio-physiological phenomena taking place inside them [66].

Table 2.3 contains a series of characteristic data of myelinated segments, experimentally determined or used in simulations.

It is worth noting the large diversity of the parameter values, both between species and within a species, but also for a single individual, between the behavior of the neurons in different areas of the central and peripheral neural system [69]. This shows the importance of identifying personalized features in diagnosis and treatment. The complexity of models requires the use of advanced optimization techniques for the experimental determination of parameter values describing each type of neuron [70].

Even with this great diversity, there are some general correlations between the parameters. Thorough statistical studies have revealed an increasing correlation between the axon diameter and the thickness of the myelin layer (proportional) but also with the inter-nodal distance (logarithmic) [71]. [72] demonstrates that these correlations are not accidental, but have an optimal character. Increasing the thickness of the myelin layer isolates the axon, lowering its linear capacitance and transverse conductance, of eddy currents. This determines a lower attenuation of the neural signal, and consequently increases the transmission length, and also the transmission speed of the signal.

The modeling of neurons and neuronal collections with high accuracy – by capturing the essential aspects of their functioning – and efficient simulation have become central issues of neuroscience. These topics are presented in monographs of theoretical neuroscience [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], in manuals of software simulators dedicated to neural systems [83, 84], or in articles describing various aspects such as: new neuronal models [85], measurement of characteristic parameters [70], computational simulation methods [86].

The diversity of neural models reflects the complexity of the underlying problem of theoretical neuroscience. The models have various forms: mathematical models described with ordinary or partial derivative equations; electrical models described as systems or electrical circuits with lumped or distributed parameters. Any of them may be linear or nonlinear. In the linear case, the equations can be represented in the frequency domain by applying suitable integral transformations (such as Fourier or Laplace), which allow the definition of transfer functions (called circuit functions in an electrical model, e.g. impedance, admittance, hybrid matrix), which greatly facilitate the study [96, 97, 72].

The complexity of a compact model is defined by its order, which is the number of state variables, respectively the number of energy accumulators (capacitors in our case) in the equivalent electric circuit. In the case of linear systems, the order corresponds to the

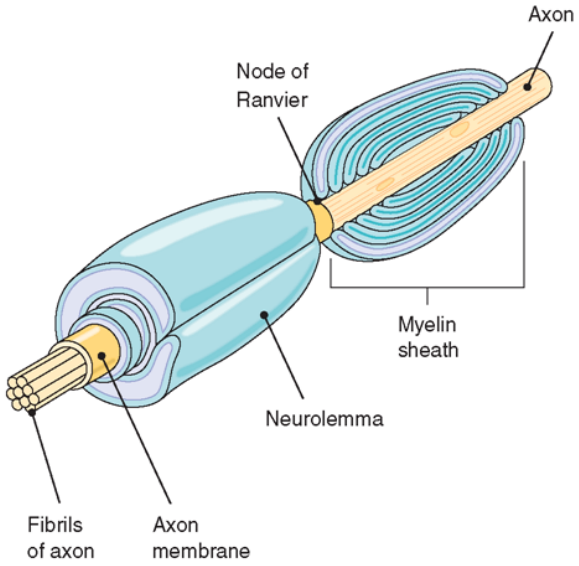


Figure 2.6: The structure of a myelinated axon [87].

Table 2.3: The characteristics of myelinated axons.

Quantity	Value	Source
Length (inter-nodal distance)	100 μm	[88]
	$250 \div 6000$ μm. Test values, typically 1500 μm	[89]
	$L/D = 100$ length vs. outer diameter	[69]
Thickness of myelin layer	$1 \div 12$ μm	[90]
	3 μm	[72]
	$d/D = 0.75(0.6 \div 0.8)$ inner-outer diameter	[69]
Axon diameter	$1 \div 20$ μm	[91]
	7 μm	[72]
	10 μm	[92, 93]
	$0.5 \div 200$ μm. Test values, typically 10 μm	[89]
	12.5 μm	[94]
Cytoplasm resistivity	100 $\Omega \cdot$ cm	[92, 93]
	0.924 $\Omega \cdot$ cm	[72]
Extracellular fluid resistivity	70 $\Omega \cdot$ cm	[93]
Myelin permittivity / capacitance	$0.005 \div 1$ μF/cm^2	[92]
	$0.005 \div 1$ μF/cm^2	[95]
	$15.44 \cdot 10^{-12}$ μF/m	[72]
Myelin conductivity	1.5 μS/cm^2 $\div$ 10.3 mS/cm^2	[92]
	$2.04 \cdot 10^{-4}$ S/m	[72]

number of poles of the transfer functions. The models with lumped parameters, described by ODEs have a finite order, whereas models with distributed parameters, described by PDEs and transcendental transfer functions, have an infinite dimension of the state space. The model is represented as a dynamic input-output system, and it can be with a single input and a single output (SISO), such as transmission models for axons and dendrites, or multiple inputs and multiple outputs (MIMO), e.g. the ones used in the study of neurons with tree-like dendrites and axons. In order to describe the behavior of a multi-terminal model, the circuits theory can be used, to derive hybrid transfer matrices whose meanings depend on the way each terminal is controlled, either in current or in voltage. If the model is linear, the system representation can be the standard state space system of equations, with the matrices: **A**, **B**, **C** and **D**. This approach is suitable since the transmission of neural signals occurs unidirectionally. Since a typical neuron has an average of 10,000 synapses, it is obvious that the order reduction of its models should be made by preserving its tree-like structure. In particular, synapses, dendrites and axons are treated as SISO-type systems, whereas soma is MISO (multiple-inputs, single-output) and the axonal tree is SIMO (single-input, multiple-outputs). Otherwise, if the neuron would be treated as a MIMO system, it would result in tens of millions of input-output connections, whose descriptions would require important computing resources, even if they had a low order [66].

The simplest models are therefore those of neuronal signal transmission through axon or dendrite sections, RC models of minimal order, having lumped parameters describing longitudinal electrical conduction phenomena through axoplasm, and capacitive and transverse conductive effects through the cell membrane. In reality, these phenomena are distributed, so the precise model is the so-called "cable model", described by 1D PDEs of parabolic type [74]. A common reduction method for these 1D models consists of segmenting the studied section into several compartments, each being minimally modeled with lumped parameters. Typically this modeling uses $5 \div 10$ compartments [92], so the global model called ”behavioral” has the same number of capacitors included in the equivalent circuit.

Another category of models takes into account the electrochemical phenomena that occur in the ion channels distributed in the cell membrane and which are active and nonlinear by excellence. The reference model in this category is Hodgkin-Huxley (HH) [98], in which one describes a linear capacitive effect, having as state quantity the membrane voltage, and the other three describe the degree of ion channels opening. Since it does not contain any spatial variables to describe a spatial distribution, we say that this model is lumped, of 0D type. The simplified modeling of the membrane has received an intense scientific attention, so that there are several nonlinear 0D models, of which the most commonly used are: FitzHugh-Nagumo (FN) [99], Frankenhaeuser-Huxley (FH) [100], Izhikevich (Iz) [101]. These models can be regarded as low-order approximations of the HH model, and are preferred in theoretical studies, precisely because of their relative simplicity [66].

For the modeling of myelinated axons, the 1D models of myelinated parts are linked with nonlinear 0D models of Ranvier nodes, resulting in non-linear PDEs [81]. This combination, in which the myelinated compartments are segmented and replaced with the

simplest model, is the standard approach currently used to simulate saltatory conduction [102], [85], [95]. This idea is implemented in most neural simulators (GENESIS, NEURON, etc.) [83], [84].

Other complicated models are reported in the literature, describing various morphological or structural details of neurons, such as the transition regions between Ranvier nodes and the axon body, called paranodes and juxtanodes [95], [93], or the layered structure of myelin [94], or the space between the neuron and this layer or between the layers [103].

Further development of these models is done by considering not only the spatial structure of neurons but also of the extracellular distribution of potential, which means the development of either 2D or 3D models. A first step is modeling the tree structures [74]. For example, in [104], the spinal neurons and their collateral deviations are studied, considering the distribution of the extracellular potential estimated with finite differences. The extracellular potential distribution is also studied in [102, 105, 73, 93, 106, 107, 108, 109, 110, 111, 112]. This distribution is important from several points of view: the interpretation of measurements made with electrodes placed in the extracellular space, the effect of partial demyelination and remyelination on the extracellular potential, the effect of stimulation by extracellular electrodes on the action potential. The problem of a non-myelinated active conducting fiber found in an infinite conducting homogeneous environment in which the scalar electric potential is harmonic is addressed in [113]. The longitudinal component of the internal and external current was determined from the solution of the Laplace equation written in cylindrical coordinates using Bessel functions integrals. The extracellular potential and the current generated by a myelinated active fiber placed in a conductive medium is computed in [93]. The distribution of transmembrane voltage is determined as a solution of the Laplace equation in cylindrical coordinates. The distribution of the potential expressed as an integral of Bessel functions results after imposing the boundary conditions [66].

By combining the tree-like axon and dendrite models with the neuronal cell body model (soma), the multi-terminal model of the singular neuron is obtained. The nonlinear dynamic behavior plays an essential role in its functioning [80], [77]. It should be noted that for the modeling of the cell body (soma), it is necessary to consider more complicated effects, including the summation of input signals, such as those described by the "integrate and fire" model [114]. This is to mention the simplest model, because in reality the neuron functioning is much more complicated, with effects specific to strongly nonlinear dynamical systems, such as stable and unstable equilibrium, bifurcations, limit cycles, which explain resting states, excitability with all-or-nothing type threshold, singular or burst pulses, refraction period, etc. The bifurcations in the phase plane describe the essence of the dynamical behavior of the neurons, which are categorized accordingly into four broad categories: integrators, resonators, monostable and bistable. A good neural model must reproduce not only its electrophysiological phenomena and especially the dynamics of its bifurcations. The phase plane gives a relevant image for model characterization. Several models with very different behaviors of neurons found in different areas of the nervous system are presented in [80].

The next major step in neural system modeling is the study of neuronal collectivities. This step involves the modeling of synapses [77], [81], [115], [116], [117]. But as the

number of simulated neurons grows, the problem becomes more and more difficult to solve, so it requires the use of a superior level of abstraction of single neuron behavior. This is why new theories were developed, such as artificial neural circuits (ANN) [118], [119] and spiking neural networks (SNN) [120], [121]. The goal is to be able to model neural systems of the complexity of those found in the human brain. Simulations of networks with 16.7 million neurons with 4 billion synapses were performed [122]. If we only refer to the myelinated compartments, their number, virtually equal to that of the Ranvier nodes, is according to [123] of 10^{11} only in the brain.

2.3 Conclusions regarding the State of the Art

This chapter presented the current state of research in systems biology in general and in neuronal modeling in particular, as they are presented in [27] and [66].

Systems biology as a holistic approach to biology explains not only the structures, but also the dynamics, control and design methods of complex biological systems. It considers the components of biological systems acting differently in isolation than if they are integrated into a larger system. Being an interdisciplinary field, systems biology is based on the advances in theory, experiments and as well in computational modeling.

The biological systems are structured in an hierarchical manner, spanning spatial scales from meters to nanometers (this means 3 times 9 magnitude orders, so 27 in 3D) and larger time-rates from years to picoseconds. The multiscale and multi-rate are therefore intrinsically correlated in the case of biological systems. Properties of such systems at a certain scale are strongly influenced by the interactions at lower and higher scales. The diversity of phenomena taking place at such different space/time scales needs several methodologies and techniques for the mathematical modeling and/or computational representations and simulation. However there are conceptual and practical gaps between scales, some of them being irreducible. They require carefully developed methods to bridge these gaps, their identification and integration being the most difficult part in the modeling process.

The hierarchical organization of biological systems may be modeled in principle, by putting together millions of cells, every cell consisting of thousands of ion channels, but it is obvious that this model is numerically intractable by nowadays computers. Consequently, there is a close connection between multiscale modeling and model reduction.

This idea is even more apparent in the particular case of neuronal systems, the human brain having around 100 billions neurons and ten times more glial cells. Without a rigorous multiscale approach, based on high performance computing techniques, in which passing from one level to another requires a careful effort to severely reduce the order of the models from the first level, it is not possible to address the problem of brain modeling [124]. It is obvious that without a careful reduction at lower levels one cannot master the complexity of real neural structures, by keeping the essence of the behavior of each element.

Chapter 3

One-dimensional Models for Neuronal Signals' Transmission

This chapter describes the 1D model of a neuronal axon as a RC transmission line (cable model). The analysis of a simple model will allow the understanding of concepts and physical phenomena as well as the validation of the numerical methods and algorithms within the modeling procedure. The chapter follows five of the six main steps of CSE modeling, according to [5] (model reduction is covered in detail in Chapter 4):

- **Conceptual modeling**: establish the geometry, materials, physical principles of functioning; decide on the simplifying hypotheses and the neglected aspects;
- **Mathematical modeling**: formulate a well defined problem in mathematical terms;
- **The (approximate) analytical modeling**: solve a simplified version of the model's equations, in order to analytically determine the relationships between the input and output quantities;
- **Numerical and computational modeling**: build a computer-based model corresponding to the mathematical model already defined;
- **Model reduction**: extract a reduced model from the numerical one, which approximates the solution of the original model with an acceptable error;
- **Verification and validation of the model**: verify the solution (compare the numerical solution with the analytical one, compare the reduced model solution with the full model solution), validate the model (compare the simulation results with the experimental results).

This chapter is the development of the ideas presented in [125].

3.1 Conceptual modeling

At this stage of modeling we describe the axon to be modeled, we analyze its principles of functioning in order to establish the geometrical model and the physical model. We

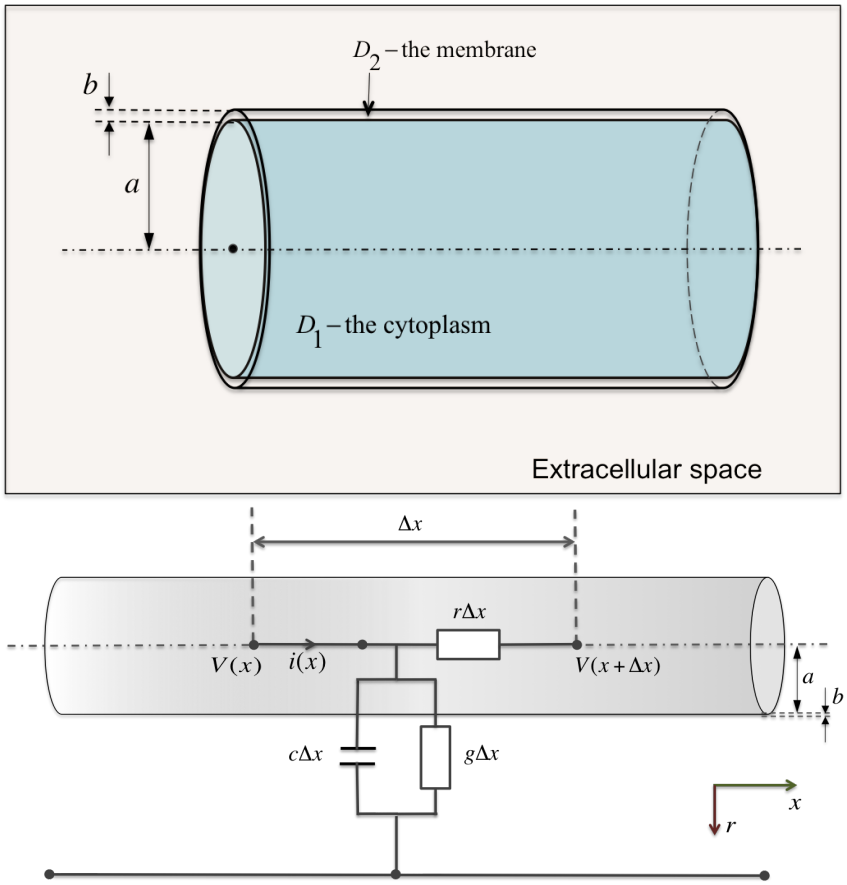


Figure 3.1: a) The spatial domains. b) Mapping with segmented line model circuit.

first identify the shape, the dimensions and the structure of the studied object. If it can be decomposed into parts then the shapes, the dimensions, the materials and the (geometric) relations between them are determined: points, edges or contact surfaces. The computing domain is set at this point, which is usually bounded by a real or fictional boundary (this must be done because most modeling methods require the field on which physical fields are defined to be finite).

3.1.1 Geometrical modeling

We consider the axon of cylindrical shape (Fig. 3.1a) and we assume that the studied system has axial and radial symmetry, that is, all physical quantities depend on a single spatial coordinate – the longitudinal one (Fig. 3.1b). Consequently, we have an axisymmetric (1.5D) computing domain, which physically is 3D and mathematically is 1D. In the cylindrical domain – D_1 – we assume that the electric potential V is dependent on the x coordinate. The axon cylinder is surrounded by its membrane – D_2.

3.1.2 Simplifying hypotheses

All the geometrical shapes are considered ideal, no roughness of the surfaces is considered. In each subdomain, the media is considered homogeneous. Any deviation from axial symmetry is neglected; therefore any angular variation of the data or of the solution

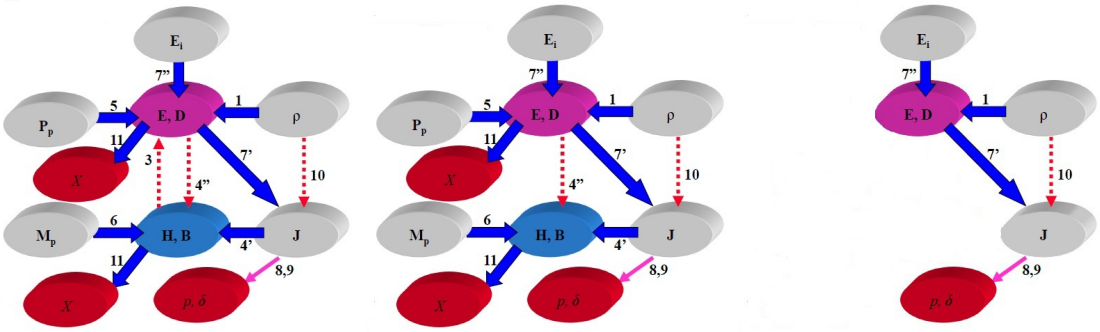


Figure 3.2: Diagram of causal connections in general variable field (left), EQS field (center) and case study (right).

is neglected accordingly. The electric field inside the membrane is considered radially oriented and with constant intensity in any cross section. Within the cytoplasm, the electric field is considered axially oriented, with no transversal variation.

3.1.3 Physical modeling

The objective of this modeling step is to make the transition from a qualitative, phenomenological, natural language description, to a quantitative representation. At this stage we identify the main physical quantities describing the state and interactions of the modeled object, the causal relations between them, thus identifying the fundamental physical phenomena based on which the object functions. The field sources and the connections between domains are emphasized. The quantities' variation over time determines the regime of functioning. As shown in Fig. 3.1b, in D_1 only conductive effects are considered, while in D_2 both resistive and capacitive effects are taken into account. Therefore, D_1 is operating in Electro Conduction region (EC), whereas D_2 is modeled with Electro Quasi Static (EQS) field. In EQS regime the bodies are immobile; the electromagnetic field varies with time, but slow enough to be able to neglect the electromagnetic induction. Formally, we can consider in equations $\mathbf{B} = 0$; there will be no quantity characteristic to the magnetic field.

The causal connections between the physical quantities in the general case (left), in the particular case – EQS regime (center) and the case study (right) are presented in Fig. 3.2.

The Maxwell's equations satisfied by the electromagnetic field in the general case (ED) and in the particular case of the EQS regime are presented in Table 3.1. The last column contains the local form of the equations in the EQS field, where the media are linear, both from dielectric and conduction points of view.

The fundamental equations of the EQS regime have the following local form [126], [127]:

Table 3.1: The Maxwell's equations satisfied by the electromagnetic field (first order equations).

		ED	EQS	Case study
1	The electric flux law	$\nabla\mathbf{D} = \rho$	$\nabla\mathbf{D} = \rho$	$\nabla\mathbf{D} = \rho$
2	The magnetic flux law	$\nabla\mathbf{B} = 0$	–	–
3	The law of electromagnetic induction (Faraday law)	$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$	$\nabla \times \mathbf{E} = 0$	$\nabla \times \mathbf{E} = 0$
4	The magnetic circuit law	$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$	–	–
5	The polarization law	$\mathbf{D} = \varepsilon\mathbf{E} + \mathbf{P}_p$	$\mathbf{D} = \varepsilon\mathbf{E} + \mathbf{P}_p$	$\mathbf{D} = \varepsilon\mathbf{E}$
6	The magnetization law	$\mathbf{B} = \mu\mathbf{H} + \mu_0\mathbf{M}_p$	–	–
7	The conduction law	$\mathbf{J} = \sigma\left(\mathbf{E} + \mathbf{E}_i\right)$	$\mathbf{J} = \sigma\left(\mathbf{E} + \mathbf{E}_i\right)$	$\mathbf{J} = \sigma\mathbf{E}$
8	The energy transfer law	$p = \mathbf{EJ}$	$p = \mathbf{EJ}$	$p = \mathbf{EJ}$
9	The mass transfer law	$\delta = k\mathbf{J}$	$\delta = k\mathbf{J}$	$\delta = k\mathbf{J}$
10	The charge conservation theorem	$\nabla\mathbf{J} = -\frac{\partial \rho}{\partial t}$	$\nabla\mathbf{J} = -\frac{\partial \rho}{\partial t}$	$\nabla\mathbf{J} = -\frac{\partial \rho}{\partial t}$

$$
\begin{aligned}
\nabla\mathbf{D} &= \rho';\\
\nabla\mathbf{J} = -\frac{\partial \rho'}{\partial t} \Rightarrow \nabla\left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}\right) &= 0;\\
\nabla \times \mathbf{E} = 0 \Rightarrow \mathbf{E} &= -\mathrm{grad}V;\\
\mathbf{D} &= \varepsilon\mathbf{E};\\
\mathbf{J} &= \sigma\left(\mathbf{E} + \mathbf{E}_i\right).
\end{aligned}
$$

Note that this differential form of equations is only valid if local quantities are derivable, hence with continuous variation in space. In D_1 the field is electro conductive (EC), and the capacitive effect becomes negligible, a condition not fulfilled in our case on the border between D_1 and D_2, which is a discontinuous surface. Here we apply either the global form of the equations (3.1) [126], [127], or the form valid on discontinuity surfaces (3.2).

$$
\begin{aligned}
\int_{\partial\Omega} \mathbf{D} \cdot \mathrm{d}\mathbf{A} = \int_{\Omega} \rho \mathrm{dv} \int_{\partial\Omega} \mathbf{J} \cdot \mathrm{d}\mathbf{A} = -\int_{\Omega} \frac{\partial \rho}{\partial t} \mathrm{dv} \Longrightarrow \int_{\partial\Omega} \mathbf{J}_t \cdot \mathrm{d}\mathbf{A} = 0\\
\Longrightarrow \int_{\partial\Omega} \mathbf{J}_t \cdot \mathrm{d}\mathbf{A} = 0; \quad \mathbf{J}_t = \mathbf{J} + \mathbf{J}_d = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t};\\
\int_{\partial S} \mathbf{E} \cdot \mathrm{d}\mathbf{r} = 0.
\end{aligned}
\tag{3.1}
$$

$$\begin{aligned}
\nabla_S \mathbf{D} = \rho_S \Leftrightarrow \mathbf{n}_{12} \cdot (\mathbf{D}_2 - \mathbf{D}_1) = \rho_S \Rightarrow u_r \cdot \mathbf{D}_2 = \rho_S \Rightarrow D_{2r} = \rho_S; \\
\nabla_S \mathbf{J} = -\frac{\partial \rho_S}{\partial t} \Leftrightarrow \mathbf{n}_{12} \cdot (\mathbf{J}_2 - \mathbf{J}_1) = -\frac{\partial \rho_S}{\partial t} \\
\Rightarrow u_r \cdot \mathbf{J}_2 = -\frac{\partial \rho_S}{\partial t} \Rightarrow J_{2r} = -\frac{\partial \rho_S}{\partial t}; \\
\nabla_S \times \mathbf{E} = 0 \Rightarrow E_{t1} = E_{t2} \Rightarrow E_{x1} = E_{x2} \Rightarrow V_1 = V_2.
\end{aligned} \tag{3.2}$$

Consequently, the solution $V(x, r, t)$ must be a continuous function on the union between D_1 and D_2. It is assumed that this function does not depend on r in D_1 and that in D_2 it decreases affinely with the radius from the inner value to zero, depending on x as in D_1. To ensure the continuity of the potential and implicitly of the tangential component of the electric field strength, this is not perfectly radial in the membrane, having a small longitudinal component that tends to zero as the radius increases. On the discontinuity surface, due to the accumulation of charges, the normal components of electric flux density and current density are not preserved, being null in D_1 and not null in D_2. In fact, these behaviors of the potential and the field in the membrane are not very relevant because they are "crushed" when the thickness of the membrane tends to zero (for unmyelinated axons), and the solution $V(x, t)$ remains dependent on just two variables, one spatial and one temporal.

If the medium is homogeneous, as each of D_1 and D_2 is respectively, we have:

$$\begin{aligned}
\varepsilon \nabla \mathbf{E} = \rho \Rightarrow -\varepsilon \Delta V = \rho; \\
\sigma \nabla \mathbf{E} = -\frac{\partial \rho}{\partial t} \Leftrightarrow \frac{\sigma \rho}{\varepsilon} = -\frac{\partial \rho}{\partial t} \Longrightarrow \rho(t) = \rho(0) e^{-\frac{t}{\tau}}; \tau = \varepsilon / \sigma; \\
\nabla \times \mathbf{E} = 0 \Rightarrow \mathbf{E} = -\operatorname{grad} V \\
\Rightarrow -\sigma \Delta V = -\frac{\partial \rho}{\partial t} = \varepsilon \Delta \frac{\partial V}{\partial t} \Longrightarrow \Delta \left(\sigma V + \varepsilon \frac{\partial V}{\partial t} \right) = 0.
\end{aligned}$$

Since the cytoplasm has a relatively good conductivity, its time constant is small, which makes its internal charge to relax practically instantaneously. Consequently, the charge in the cytoplasm has null volume density and tends to be superficially distributed. This explains why in D_1 the field is electro-conductive, and the capacitive effect becomes negligible. Instead, inside the membrane things are different, here both capacitive and conductive effects are important. The membrane is traversed by a total current, which has a resistive component (conduction current due to losses) and a capacitive one (the displacement current):

$$\mathbf{J}_t = \mathbf{J} + \mathbf{J}_d = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} = \sigma \mathbf{E} + \varepsilon \frac{\partial \mathbf{E}}{\partial t}.$$

These current densities are radially oriented and their sum is preserved (it is solenoidal, meaning it has a null divergence).

Table 3.2: The material parameters used in the case study.

Parameter	Computed value
The dielectric permeability of the membrane	$\varepsilon = 15.44 \cdot \varepsilon_0 = 15.44 \cdot 8.85 \cdot 10^{-12}$ F/m
The membrane conductivity	$\sigma = 2.04 \cdot 10^{-4}$ S/m
The resistivity of the cytoplasm	$\rho = 0.924\ \Omega \cdot$ m

Table 3.3: The line parameters used in the case study.

Parameter	Expression	Numeric value
Line capacity	$c = \frac{C}{L} = \frac{\varepsilon\pi 2a}{b}$	$c = 2 \cdot 10^{-12}$ F/mm = 2 pF/mm
Line resistance	$r = \frac{R}{L} = \frac{\rho}{\pi a^2}$	$r = 6 \cdot 10^6\ \Omega$/mm = 6 MΩ/mm
Line conductance of the membrane	$g = \frac{G}{L} = \frac{\sigma\pi 2a}{b}$	$g = 0.003$ mS/mm

3.2 Mathematical modeling

The mathematical modeling aims to formulate the field analysis problem in mathematical terms and verify that the problem is well formulated.

3.2.1 Problem data

The geometry of the computing domain

$a = 7\ \mu$m – axon's radius (inner diameter $d = 2a$, outer diameter $D = 2(a + b)$)

$b = 3\ \mu$m – membrane thickness

Material constants

In the literature there are various values for material parameters, sometimes varying with the frequency [128], [129], [130].

Timotin considered in his study [72] the line parameters r, g and c used by FitzHugh [102] and previously measured by Tasaki [131]. Table 3.2 contains the computation of the material parameters for which the line parameters have the values used by Timotin and by us in the case study.

Note: from this moment on we will denote by ρ the resistivity (in the previous section this symbol was used for the charge density).

Using these geometrical and material data, the line parameters' values can be computed, the values are contained in Table 3.3.

3.2.2 The problem unknowns and the equation to be solved

The main unknown is the electric potential distribution $V(x,t)$ in the computing domain $0 < x < L$, at any moment in time $0 < t < T$. The equations are those of the linear 1D model (RC transmission line). According to Kirchhoff's relation (deduced from the charge conservation theorem) written on the line segment in Fig. 3.1b, we have:

$$i(x) = i(x + \Delta x) + g\Delta x V(x,t) + c\Delta x \frac{\partial V(x,t)}{\partial t}.$$

Dividing by Δx we obtain:

$$\frac{i(x + \Delta x) - i(x)}{\Delta x} = gV(x,t) + c\frac{\partial \mathrm{V(x,t)}}{\partial \mathrm{t}},$$

and then when $\Delta x \to 0$:

$$-\frac{\partial i(x,t)}{\partial x} = gV(x,t) + c\frac{\partial \mathrm{V(x,t)}}{\partial \mathrm{t}}. \tag{3.3}$$

On the other hand, according to Kirchhoff II relation (consequence of the potential theorem), it follows that:

$$V(x) - V(x + \Delta x) = i(x + \Delta x) r \Delta x.$$

If we divide this relation by Δx, having this elementary length tend to 0, we obtain a second relation:

$$-\frac{\partial V(x,t)}{\partial x} = ri(x,t). \tag{3.4}$$

The relations (3.3) and (3.4) represent the first order equations of the RC transmission lines. After derivation by x and using (3.3), the current is eliminated and the second order equation satisfied by the potential is obtained:

$$\frac{\partial^2 V(x,t)}{\partial x^2} = rgV(x,t) + rc\frac{\partial V(x,t)}{\partial t}, \tag{3.5}$$

form that allows the writing using relative, dimensionless spatial and temporal variables. The relation (3.5) is a linear equation with partial derivatives of parabolic type, which describes the diffusion phenomenon (it is also called the heat equation). It is the RC transmission line equation, which represents the foundation of the neural model in "cable theory" [132], [133], [134].

Having defined the time constant $\tau = \frac{c}{g} = 0.6$ μs and the length constant $\lambda_0 = \frac{1}{\sqrt{rg}} =$ 223 μm, (3.5) is rewritten as:

$$\lambda_0^2 \frac{\partial^2 V(x,t)}{\partial x^2} = V(x,t) + \tau \frac{\partial V(x,t)}{\partial t}. \tag{3.6}$$

The time constant indicates the time at which the initial condition is attenuated e times, having the boundary conditions null and the space constant indicates the distance at which the step signal applied at the initial point ($x = 0$) is attenuated e times.

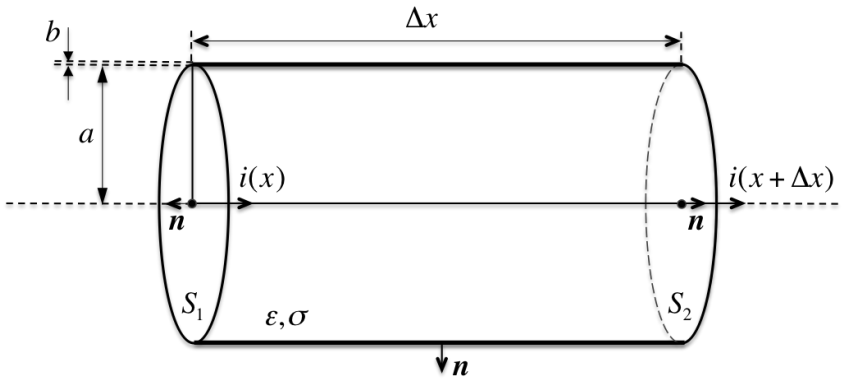


Figure 3.3: The charge conservation theorem applied on two cross sections.

The relations (3.3) and (3.4) and the manner in which they were obtained show that the studied system is a distributed parameter circuit. It is very important to note that (3.3) and (3.4) can be obtained directly from the EQS field equations, without the need to use Kirchhoff's relations. This alternative approach has the advantage of emphasizing the simplification assumptions of physical modeling, and it is more natural to be generalized for 2D and 3D geometric models.

In order to obtain (3.3), we will consider two cross sections S_1 and S_2, spaced by Δx (Fig. 3.3) and we will apply on the cylinder with this height the global form of the charge conservation theorem:

$$\begin{aligned}\int_{\partial\Omega} \mathbf{J}_t \cdot \mathbf{dA} = 0 &\Longrightarrow \int_{\partial\Omega} \left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}\right) \cdot \mathrm{d}\mathbf{A} = 0 \\ &\Longrightarrow \int_{S_1} \mathbf{J} \cdot \mathrm{d}\mathbf{A} + \int_{S_2} \mathbf{J} \cdot \mathrm{d}\mathbf{A} + \int_{S_l} \mathbf{J} \cdot \mathrm{d}\mathbf{A} + \int_{S_l} \frac{\partial \mathbf{D}}{\partial t} \cdot \mathrm{d}\mathbf{A} = \mathbf{0} \\ &\Longrightarrow -i(x) + i(x+\Delta x) + 2\pi a \Delta x \sigma E + 2\pi a \Delta x \varepsilon \frac{\partial E}{\partial t} = 0\end{aligned}$$

$$\Longrightarrow gV(x,t) + c\frac{\partial V}{\partial t} = -\frac{\partial i}{\partial x}, \tag{3.7}$$

with $g = \frac{\sigma 2\pi a}{b}$, $c = \frac{\varepsilon 2\pi a}{b}$, where $E = V(x,t)/b$ is the field strength in the membrane.

The relation (3.4) is a consequence of the potential theorem:

$$\nabla \times \mathbf{E} = 0 \Rightarrow \mathbf{E} = -\operatorname{grad} V = \rho \mathbf{I} \Rightarrow -\frac{\partial V}{\partial x} = \frac{\rho i}{\pi a^2}$$

$$\Leftrightarrow -\frac{\partial V}{\partial x} = ri; \quad r = \frac{\rho}{\pi a^2} \tag{3.8}$$

Note that this approach allows not only the proof of transmission line equations, but also the extraction of line parameters' expressions. Considering the problem domain as the reunion between D_1 and D_2, on the outer surface the boundary condition is Dirichlet with $V = 0$. If the thickness of the membrane b is negligible, we can assume that the domain of our problem is only D_1, and in this case the boundary condition on the outer

surface $r = a$ is a superficial admittance BC.

$$\begin{aligned} -J_x(x)\pi a^2 + J_x(x+\Delta x)\pi a^2 + 2\pi a \Delta x J_{rt} &= 0 \\ \Leftrightarrow \operatorname{div} \mathbf{J}_1 + \frac{2}{a} \operatorname{div}_s \mathbf{J}_{2t} &= 0 \\ \Rightarrow \frac{\partial J_x}{\partial x} + \frac{2}{a} J_{rt} &= 0 \\ \Rightarrow \operatorname{div} \mathbf{J}_1 = -\frac{2}{a} \operatorname{div}_s J_{2t} = -\frac{2}{a} \left(\sigma \frac{V}{b} + \frac{\varepsilon}{b} \frac{\partial V}{\partial t} \right) & \end{aligned}$$

In reality, the computing domain of the problem is a two-dimensional rectangle: $0 < x < L$; $0 < t < T$, the solution $V(x,t)$ of equation (3.5) being defined on this domain. Boundary conditions must be added to the equation, in order for the problem to be well formulated. The BCs are described in the following section.

3.2.3 Boundary conditions

The boundary conditions are set as follows:

– At the left side of the cylinder ($x = 0$): non-null Dirichlet conditions $V = e(t)$, where $e(t)$ is a known excitation signal (standard). For the standard neural signal the following approximate expression is used [135], [136]:

$$V(0,t) = e(t) = V_0 + V_m \left(e^{-t/\tau_1} - e^{-t/\tau_2} \right) \tag{3.9}$$

The numerical values are:

$$\begin{aligned} V_0 &= -70 \text{ mV}; \\ V_m &= 700 \text{ mV}; \\ \tau_1 &= 0.3 \cdot 10^{-3} \text{ s}; \\ \tau_2 &= 0.2 \cdot 10^{-3} \text{ s}. \end{aligned}$$

This input signal is represented graphically in Fig. 3.4.

– At the right side of the cylinder ($x = L$): null Neumann BCs:

$$\frac{\partial V}{\partial x} = 0, \tag{3.10}$$

which correspond to an open-circuited line (negligible output current).

3.2.4 Initial conditions

The distribution of the electric potential at the initial time $t = 0$ corresponds to the equilibrium value, when no signal is traveling [137], [138]:

$$V(x,0) = V_0 = -70 \text{ mV} \tag{3.11}$$

for the entire spatial domain $0 < x < L$.

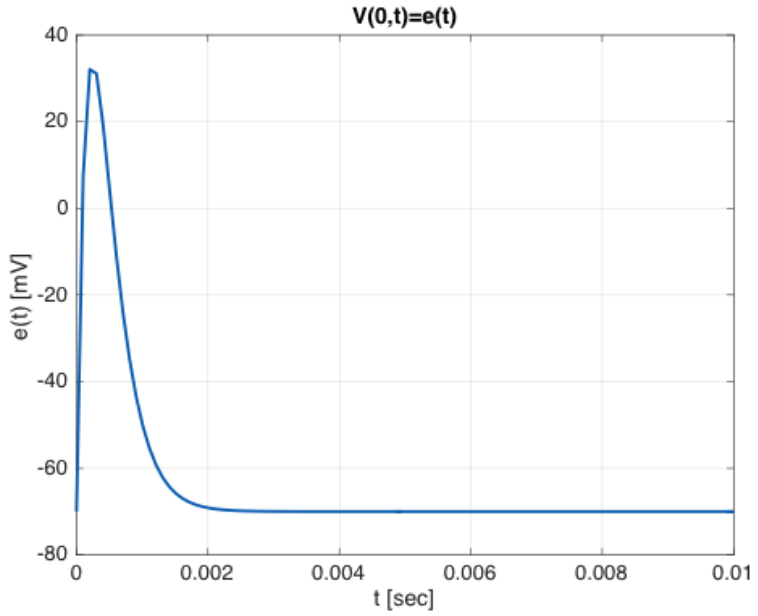


Figure 3.4: The standard excitation signal $V(0,t) = e(t)$.

3.3 The analytical solution

We are dealing with a linear model, so the problem can be also solved with analytical methods. The analytical solution of the partial differential equation will be used to validate the numerical methods. This validation is timely, as the nonlinear models to be analyzed later – describing Ranvier nodes and ion channels – can only be solved numerically, so we need reliable numerical procedures.

3.3.1 The operational solution

The solution can be determined using the Laplace transforms:

$$V(0,t) = e(t) = V_0 + V_m\left(e^{-t/\tau_1} - e^{-t/\tau_2}\right)$$
$$\xrightarrow{L} \mathrm{E}(s) = \frac{V_0}{s} + V_m\left(\frac{1}{s+1/\tau_1} - \frac{1}{s+1/\tau_2}\right).$$

The equation (3.5) has the following operational form:

$$\frac{\mathrm{d}^2V(x,s)}{\mathrm{d}x^2} = rgV(x,s) + rcsV(x,s) - rcV_0$$
$$\Longrightarrow \frac{\mathrm{d}^2V(x,s)}{\mathrm{d}x^2} = r(g+cs)V(x,s) - rcV_0$$
$$\Longrightarrow V(x,s) = \frac{cV_0}{g+sc} + \frac{1}{r(g+sc)}\frac{\mathrm{d}^2V(x,s)}{\mathrm{d}x^2}$$

This is a second order ordinary differential equation, with constant coefficients and the independent variable x, having the characteristic equation:

$$1 = \frac{\lambda^2}{r(g+sc)},$$

with the roots $+/-\lambda$ where:

$$\lambda = \sqrt{r(g+sc)}.$$

The general solution is:

$$V(x,s) = \frac{cV_0}{g+sc} + C_1 \operatorname{sh}(\lambda x) + C_2 \operatorname{sh}(\lambda(x-L)).$$

It is verified that:

$$\begin{aligned} C_1 \operatorname{sh}(\lambda x) + C_2 \operatorname{sh}(\lambda(x-L)) &= \frac{1}{r(g+sc)} \left(r(g+sc)V(x,s) - rcV_0\right) \\ \Longrightarrow C_1 \operatorname{sh}(\lambda x) + C_2 \operatorname{sh}(\lambda(x-L)) &= V(x,s) - \frac{V_0}{s+g/c}. \end{aligned}$$

The boundary conditions at $x = 0$ and $x = L$ are used to determine the two constants C_1 and C_2:

$$\begin{aligned} x = 0 \Longrightarrow \mathrm{V}(0,s) &= \frac{V_0}{s+\frac{g}{c}} - C_2 \operatorname{sh}(\lambda L) = \mathrm{E}(s) \\ \Longrightarrow C_2 &= \frac{-E(s) + \frac{V_0}{s+g/c}}{\operatorname{sh}(\lambda L)}; \\ x = L \Longrightarrow \frac{\mathrm{d}V}{\mathrm{d}x} &= C_1 \lambda \operatorname{ch}(\lambda x) + C_2 \lambda \operatorname{ch}(\lambda(x-L)) = C_1 \lambda \operatorname{ch}(\lambda L) + C_2 \lambda = 0 \\ \Longrightarrow C_1 &= -C_2 / \operatorname{ch}(\lambda L). \end{aligned}$$

$$\begin{aligned} \mathrm{V}(x,s) &= \frac{V_0}{s+g/c} + \frac{-E(s) + \frac{V_0}{s+g/c}}{\operatorname{sh}(\lambda L)} \left(-\frac{1}{\operatorname{ch}(\lambda L)} \operatorname{sh}(\lambda x) + \operatorname{sh}(\lambda(x-L)) \right) \\ \mathrm{V}(L,s) &= \frac{V_0}{s+g/c} + \frac{1}{\operatorname{ch}(\lambda L)} \left(\mathrm{E}(s) - \frac{V_0}{s+g/c} \right) \\ \Longrightarrow V(L,s) &= \frac{\mathrm{E}(s)}{\operatorname{ch}(\lambda L)} + \frac{V_0(1 - 1/\operatorname{ch}(\lambda L))}{s+g/c}. \end{aligned}$$

$$\begin{aligned} &t \to 0 \Rightarrow s \to \infty; \lambda \to \infty; V(x,0) = \lim_{s\to\infty} s\mathrm{V}(L,s) \to V_0; \\ &t \to \infty \Rightarrow s \to 0; \lambda \to \sqrt{rg} = 1/\lambda_0; V(x,\infty) = \lim_{s\to 0} s\mathrm{V}(L,s) \to V_0/\operatorname{ch}(L\sqrt{rg}); \\ &\mathrm{V}(L,s) = \frac{V_0}{s \cdot \operatorname{ch}(\lambda L)} + \frac{V_m}{\operatorname{ch}(\lambda L)} \left(\frac{1}{s+1/\tau_1} - \frac{1}{s+1/\tau_2} \right) + \frac{V_0}{s+g/c} - \frac{V_0}{\operatorname{ch}(\lambda L)(s+g/c)}, \end{aligned}$$

with $\lambda = \sqrt{r(g+sc)}$, therefore:

$$\mathrm{V}(L,s) = \frac{1}{\operatorname{ch}(\lambda L)} \left(\frac{V_0}{s} + \frac{V_m}{s+1/\tau_1} - \frac{V_m}{s+1/\tau_2} - \frac{V_0}{s+g/c} \right) + \frac{V_0}{s+g/c}. \tag{3.12}$$

The term $\frac{1}{\operatorname{ch}(\lambda L)}$ makes it harder to shift into the time domain, because $\operatorname{ch}(\lambda L) = 0$ has

an infinity of solutions $\lambda L = \frac{1}{2} j\pi(2n-1), n \in \mathbb{Z}$, with the corresponding time constants:

$$\begin{aligned} \lambda L = \tfrac{j\pi(2n-1)}{2} \quad &\Longrightarrow \lambda^2 L^2 = -\left(\tfrac{\pi(2n-1)}{2}\right)^2 \\ &\Longrightarrow r(g+sc) = -\left(\tfrac{\pi(2n-1)}{2L}\right)^2 \\ &\Longrightarrow sc = -\frac{\left(\frac{\pi(2n-1)}{2L}\right)^2}{r} - g \\ &\Longrightarrow \tau_n = -\tfrac{1}{s} = \frac{c}{\frac{\left(\frac{\pi(2n-1)}{2L}\right)^2}{r}} + g. \end{aligned}$$

The largest of them is the first one:

$$\tau_1 = \frac{c}{\frac{\left(\frac{\pi}{2L}\right)^2}{r} + g}.$$

The transfer function of the studied system is:

$$\mathrm{F}(s) = \left.\frac{\mathrm{V}(L,s)}{\mathrm{E}(s)}\right|_{V_0=0} = \frac{1}{\mathrm{ch}(\lambda L)}, \tag{3.13}$$

and the effect of the initial condition is described by the transfer function:

$$\mathrm{G}(s) = \left.\frac{\mathrm{V}(L,s)}{V_0}\right|_{\mathrm{E}(s)=0} = \frac{1 - 1/\mathrm{ch}(\lambda L)}{s + g/c}.$$

3.3.2 The transfer function of the system: $\mathrm{F}(s)$

The transfer function $\mathrm{F}(s) = \left.\frac{\mathrm{V}(L,s)}{\mathrm{E}(s)}\right|_{V_0=0} = \frac{1}{\mathrm{ch}(\lambda L)}$ has no zeros and an infinity of poles, all of them real:

$$s_{F\infty} = -\frac{1}{rc}\left(\frac{\pi(2n-1)}{2L}\right)^2 - \frac{g}{c}. \tag{3.14}$$

All the values r, c, and g are positive, so all the poles are negative.

The effect of the line length L

To represent the Bode diagram, a series of nodes in the frequency domain was considered and the system response was computed (the value of the transfer function having $s = j\omega$) in those points:

freq=[0,100,1000,5000,10000,20000,1e5,2e5,5e5,1e6,2e6,3e6,4e6,5e6] Hz.

Fig. 3.5 represents the Bode diagram of $F(s)$ for three values of the line length L:

a) $L = 0.25\lambda_0 = 0.25 \cdot 223\ \mu\mathrm{m} = 0.0558$ mm

b) $L = \lambda_0 = 223\ \mu\mathrm{m} = 0.223$ mm

c) $L = 4\lambda_0 = 4 \cdot 223\ \mu\mathrm{m} = 0.892$ mm

An important characteristic parameter is represented by the value in origin of the transfer function $F(0) = \frac{1}{\mathrm{ch}(L\sqrt{rg})} = \frac{1}{\mathrm{ch}(L/\lambda_0)}$ with $\lambda_0 = \frac{1}{\sqrt{rg}}$ the characteristic length. This describes the stationary attenuation of the output signal against the input signal. The

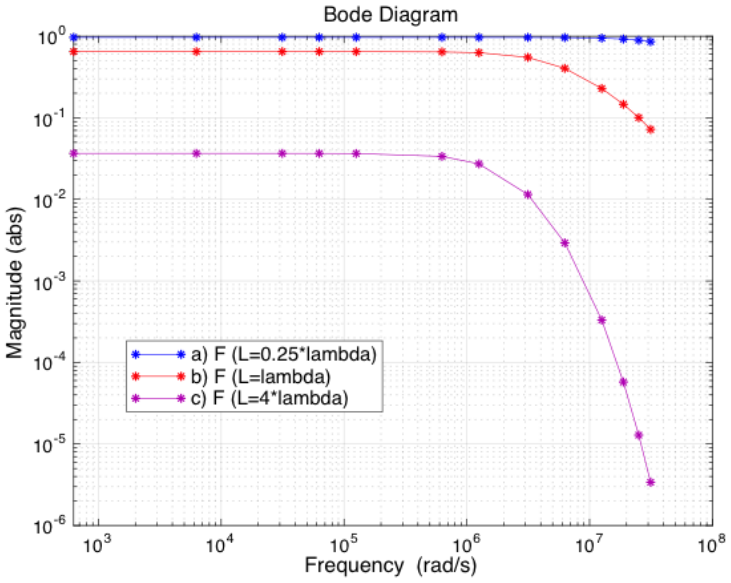


Figure 3.5: The Bode diagrams of F($j\omega$) for 3 line lengths.

denominator expression has the asymptotic behavior $\mathrm{ch}\,(L/\lambda_0) = \frac{e^{L/\lambda_0 - e^{-L/\lambda_0}}}{2} \to \frac{e^{L/\lambda_0}}{2}$, for $L \gg \lambda_0$, with an exponential growth with the line length. Consequently, the input signal is almost completely attenuated, if the line length is much larger than the characteristic length λ_0.

From the Bode diagrams it can be deducted that the signal attenuation is similar to the stationary attenuation, as long as the signal is slowly variable over time.

If the signal frequency exceeds the value of $f_1 = \frac{1}{\tau_1} = \left(\frac{\pi}{2L}\right)^2 \frac{1}{rc} + \frac{g}{c} \approx 1.51 \cdot 10^6$, corresponding to the first time constant, the magnitude decreases with 20 dB in a decade, following that at high frequencies, much higher than that corresponding to the 3$^{\text{rd}}$-4$^{\text{th}}$ time constant, the output signal to be negligible to the input signal. The line therefore has a low-pass filter behavior of superior order.

The conclusion is that the modeling system transmits the signal from input to output without attenuation if the line length is below the characteristic length and the signal frequency is below the frequency corresponding to the first time constant. Outside this scope, the transmitted signal is attenuated and dispersed (changes shape, the impulse having the tendency to widen), the farther away we are from the undisturbed transmission range.

The standard neural signal has a time constant of a millisecond, which is much larger than τ_1, so we can say that this signal is not attenuated due to its dynamic character, but because of the line length. As the threshold level is about 1/4 of the amplitude of the signal, it follows that signals that are four times attenuated will not trigger the action potential once they have reached the end of the line.

This happens when $e^{L/\lambda_0} = 8$, so for $L = 2\lambda_0$, meaning for L of about 0.5 mm. We can state that axons with geometric and material characteristics as in this case study will not be able to transmit neural signals over distances greater than 0.5 mm if they do not

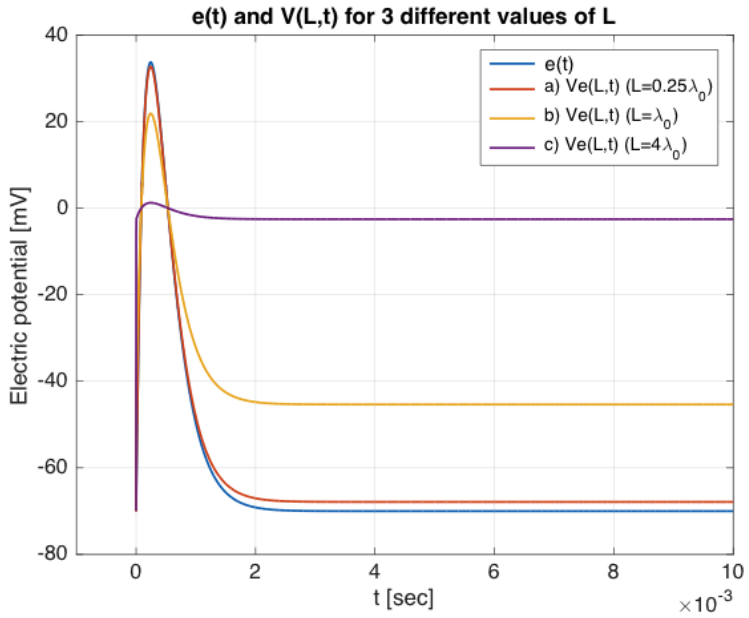


Figure 3.6: The input signal and the exact response $V(L,t)$ for 3 line lengths, around the length constant $\lambda_0 = \frac{1}{\sqrt{rg}} = 223\ \mu$m.

have ion channels to recondition these signals.

3.3.3 Transition to time domain

In the operational domain we have:

$$\mathrm{V}(L,s) = \frac{V_0}{s \cdot \mathrm{ch}(\lambda L)} + \frac{V_m}{\mathrm{ch}(\lambda L)} \left(\frac{1}{s+1/\tau_1} - \frac{1}{s+1/\tau_2} \right) + \frac{V_0}{s+g/c} - \frac{V_0}{\mathrm{ch}(\lambda L)(s+g/c)}.$$

The response $V(L,t)$ corresponding to this operational representation was determined in MATLAB using the procedure INVLAP [139]. In Fig. 3.6 this response is represented for three line lengths L, around the length constant $\lambda_0 = \frac{1}{\sqrt{rg}} = 223\ \mu$m.

For a line length shorter than the length constant ($L = 0.25\lambda_0 < \lambda_0$), the time response is close to the input, this is apparent in Fig. 3.5 (lines blue and red).

The deviation between input and output is computed using the Euclidean norm as:

$$\mathrm{err}_{\mathrm{abs}} = \|e - V_e\| = \sqrt{\frac{\sum_k |e_k - V_{e_k}|^2}{n}},$$

with n the size of the array e, and

$$\mathrm{err}_{\mathrm{rel}} = \frac{\mathrm{err}_{\mathrm{abs}}}{|\max(e)|}.$$

The absolute and relative errors for $L = 0.25\lambda_0$ have the values:

$$\mathrm{err}_{\mathrm{abs}} = 2.02;$$
$$\mathrm{err}_{\mathrm{rel}} = 0.06.$$

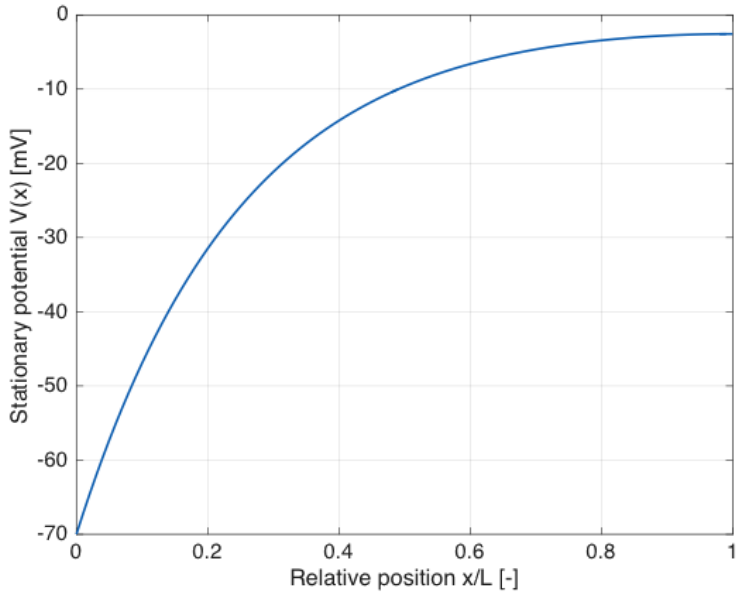


Figure 3.7: The electric potential distribution along the line, in post-transient stationary mode.

The output signal represented in Fig. 3.5 starts from equilibrium value V_0, increases to the maximum value and then decreases to the asymptotic value of:

$$V(L,t) \to \frac{V_0}{\operatorname{ch}\left(\frac{L}{\lambda_0}\right)} \approx 2V_0 e^{-\frac{L}{\lambda_0}}, \text{ for } L \gg \lambda_0,$$

which is smaller as the length L is larger. In other words, in the case of relatively long lines, we find that in our model, after the impulse, the output signal in the stationary mode following the transient mode tends to zero, and not to the equilibrium value V_0, as is actually the case when the ion pumps reestablish the balance. In our model, the electrical potential has an increasing variation along the line from $V_0 = -70$ mV to $V_0/\operatorname{ch}(L/\lambda_0) \approx 0$ with the expression:

$$\begin{aligned} s\mathrm{V}(x,s) &= \frac{sV_0}{s+g/c} + \frac{-E(s)+\frac{V_0}{s+g/c}}{\operatorname{sh}(\lambda L)}\left(-\frac{1}{\operatorname{ch}(\lambda L)}\operatorname{sh}(\lambda x)+\operatorname{sh}(\lambda(x-L))\right); \\ &\overset{s\to 0}{\to} V(x,\infty) = -\frac{V_0}{\operatorname{sh}(\lambda L)}\left(-\frac{1}{\operatorname{ch}(\lambda L)}\operatorname{sh}(\lambda x)+\operatorname{sh}(\lambda(x-L))\right). \end{aligned}$$

When $t \to \infty \Rightarrow s \to 0; \lambda \to \sqrt{rg}; V(x,\infty) = \lim_{s\to 0} s\mathrm{V}(x,s); V(L,t) \to V_0/\operatorname{ch}(L\sqrt{rg})$.

The distribution of potential along the line in the stationary mode $V(x,\infty)$ is represented in Fig. 3.7.

3.3.4 Modeling the ion pumps

The observed behavior is obviously a modeling deficiency that can be corrected if we consider the effect of the ion pumps, adding in Fig. 3.1b – in series with the transversal resistor – an ideal voltage source $E = V_0$ (Fig. 3.8).

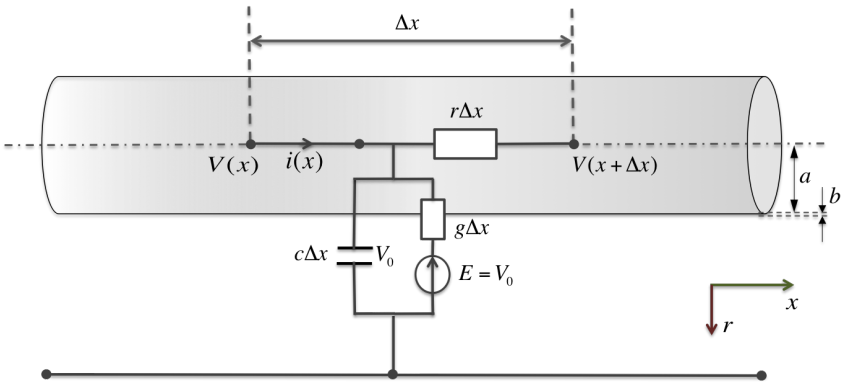


Figure 3.8: The new model which takes into account the effect of ion pumps [125].

In these conditions the current through the source is:

$$V(x,t) = E + \frac{i'}{g\Delta x} \Rightarrow i' = (V(x,t) - V_0)\, g\Delta x.$$

According to Kirchhoff' relation (resulted from the charge conservation theorem) written on the line segment, it follows that:

$$i(x) = i(x+\Delta x) + g\Delta x\,(V(x,t) - V_0) + c\Delta x \frac{\partial V(x,t)}{\partial t}.$$

Dividing by Δx we obtain:

$$\frac{i(x+\Delta x) - i(x)}{\Delta x} = g\,(V(x,t) - V_0) + c\frac{\partial \mathrm{V(x,t)}}{\partial \mathrm{t}},$$

and having $\Delta x \to 0$, then we have:

$$-\frac{\partial i(x,t)}{\partial x} = g\,(V(x,t) - V_0) + c\frac{\partial \mathrm{V(x,t)}}{\partial \mathrm{t}}, \tag{3.15}$$

which replaces (3.3). From Kirchhoff II, it results:

$$\frac{\partial V(x,t)}{\partial x} = ri(x,t). \tag{3.16}$$

These relations lead to the second order equation satisfied by the potential:

$$\frac{\partial^2 V(x,t)}{\partial x^2} = rg\,(V(x,t) - V_0) + rc\frac{\partial V(x,t)}{\partial t}, \tag{3.17}$$

or equivalently:

$$\lambda_0^2 \frac{\partial^2 V(x,t)}{\partial x^2} = V(x,t) - V_0 + \tau \frac{\partial V(x,t)}{\partial t}, \tag{3.18}$$

with the time constant $\tau = c/g = 0.6\ \mu$s and length constant $\lambda_0 = \frac{1}{\sqrt{rg}} = 223\ \mu$m and $V_0 = -70$ mV.

The equation (3.17) has the following operational form:

$$\begin{aligned}
\frac{\partial^2 \mathrm{V}(x,s)}{\mathrm{d}x^2} &= rg\mathrm{V}(x,s) + rcs\mathrm{V}(x,s) - r\left(c + \frac{g}{s}\right) V_0 \\
&\Longrightarrow \frac{\mathrm{d}^2 \mathrm{V}(x,s)}{\mathrm{d}x^2} = r(g+cs)\mathrm{V}(x,s) - r\left(c + \frac{g}{s}\right) V_0 \\
&\Longrightarrow \mathrm{V}(x,s) = \frac{cV_0}{s} + \frac{1}{r(g+sc)} \frac{\mathrm{d}^2 V(x,s)}{\mathrm{d}x^2}.
\end{aligned}$$

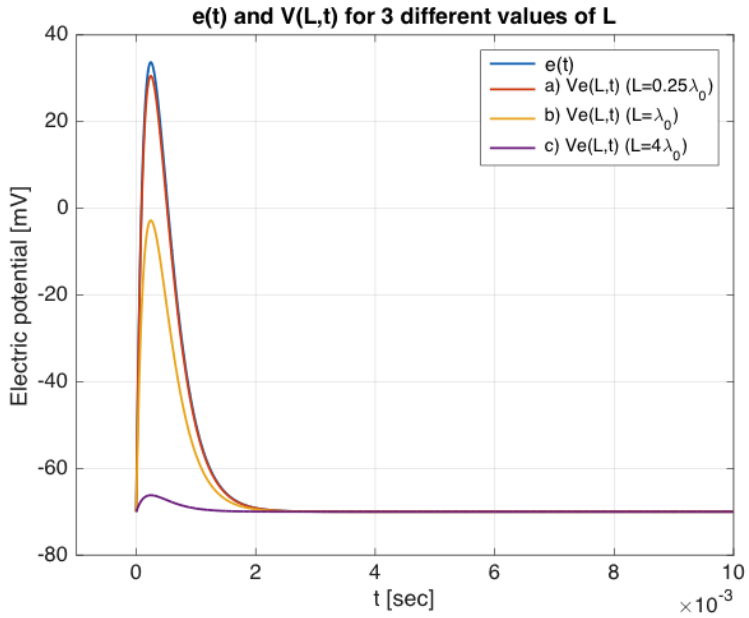


Figure 3.9: The input signal and the exact response $V(L,t)$ for 3 line lengths, around the length constant $\lambda_0 = \frac{1}{\sqrt{rg}} = 223\ \mu$m, for the model with ion pumps.

This second order ordinary differential equation with constant coefficients has the general solution of the form deducted below (to which a particular solution of the shape of the free term is added).

$$
\begin{aligned}
&C_1 \operatorname{sh}(\lambda x) + C_2 \operatorname{sh}(\lambda(x-L)) = V(x,s) - \frac{V_0}{s};\\
&\quad x = 0 \Longrightarrow V(0,s) = \frac{V_0}{s} - C_2 \operatorname{sh}(\lambda L) = \mathrm{E}(s)\\
&\qquad \Longrightarrow C_2 = \frac{-\mathrm{E}(s) + \frac{V_0}{s}}{\operatorname{sh}(\lambda L)};\\
&\quad x = L \Longrightarrow \frac{\mathrm{d}V}{\mathrm{d}x} = C_1\lambda \operatorname{ch}(\lambda x) + C_2\lambda \operatorname{ch}(\lambda(x-L)) = C_1\lambda \operatorname{ch}(\lambda L) + C_2\lambda = 0\\
&\qquad \Longrightarrow C_1 = -C_2/\operatorname{ch}(\lambda L).
\end{aligned}
$$

$$
\mathrm{V}(x,s) = \frac{V_0}{s} + \frac{-E(s) + \frac{V_0}{s}}{\operatorname{sh}(\lambda L)} \left(-\frac{1}{\operatorname{ch}(\lambda L)} \operatorname{sh}(\lambda x) + \operatorname{sh}(\lambda(x-L)) \right); \tag{3.19}
$$

$$
\mathrm{V}(L,s) = \frac{V_0}{s} + \frac{1}{\operatorname{ch}(\lambda L)} \left(\mathrm{E}(s) - \frac{V_0}{s} \right) = \frac{\mathrm{E}(s)}{\operatorname{ch}(\lambda L)} + \frac{V_0(1 - 1/\operatorname{ch}(\lambda L))}{s}. \tag{3.20}
$$

$$
\begin{aligned}
&t \to 0 \Rightarrow s \to \infty; \lambda \to \infty; V(L,0) = \lim_{s\to\infty} sV(L,s) \to V_0;\\
&t \to \infty \Rightarrow s \to 0; \lambda \to \sqrt{rg} = 1/\lambda_0; V(L,\infty) = \lim_{s\to 0} sV(L,s) \to V_0.
\end{aligned}
$$

This time the signals have the asymptotic values as expected. The variation of $V(L,t)$ over time for three line lengths is represented in Fig. 3.9.

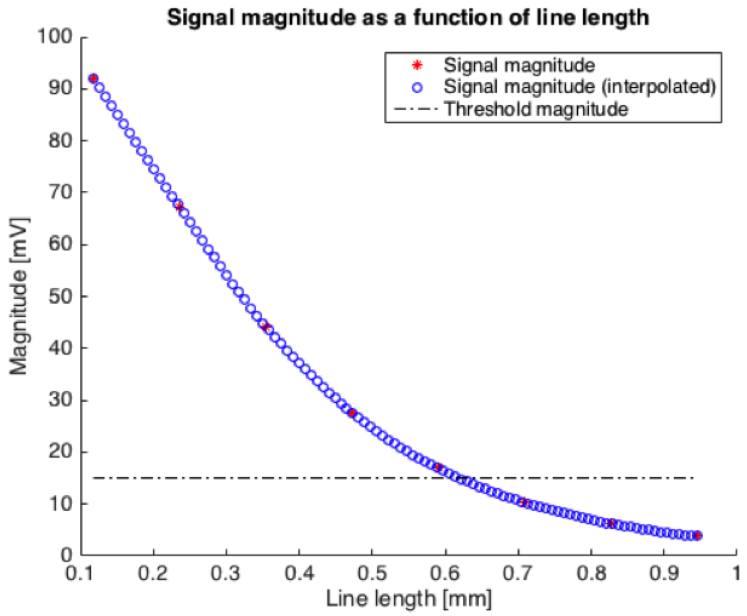


Figure 3.10: The signal amplitude vs. line length.

This model, which includes the effect of ion pumps, is present in the literature [140], [141], although the standard cable model does not contain this correction [132], [133], [134], [142].

3.3.5 Characteristic parameters: line attenuation and maximum transmission length

Numerical approach

We compute **the maximum line length** for which the signal amplitude reaches the threshold $V_t = -55$ mV (typical value). We start by representing the signal amplitude as a function of the line length (Fig. 3.10). For line lengths below the characteristic length, the signal amplitude ($V_{max} - V_{min}$) is close to the amplitude of the input signal (100 mV = (30 – (– 70)) mV). As the line length increases, the amplitude decreases so that over a certain line length, the amplitude is no longer enough to reach the threshold potential ((– 55 – (– 70)) mV = 15 mV) and trigger an action potential.

The signal magnitude was computed using INVLAP with the relation (3.20) for eight line lengths between $0.5\lambda_0$ and $4\lambda_0$ and the results were interpolated for 100 intervals (Fig. 3.10).

The line length corresponding to the threshold is $L_{\max} = 0.6136$ mm, meaning $2.6\lambda_0$. Fig. 3.11 shows the variation of $V(L,t)$ over time for four line lengths, including $L_{\max}$.

Analytical approach

As stated earlier, one of the characteristic parameters is **the attenuation factor of the stationary line**, defined by the value in origin of the transfer function:

$$F(0) = \frac{1}{\mathrm{ch}(L\sqrt{rg})} = \frac{1}{\mathrm{ch}\,(L/\lambda_0)} \cong \frac{e^{-L/\lambda_0}}{2}, \text{ with } \lambda_0 = \frac{1}{\sqrt{rg}} \text{ the characteristic length.}$$

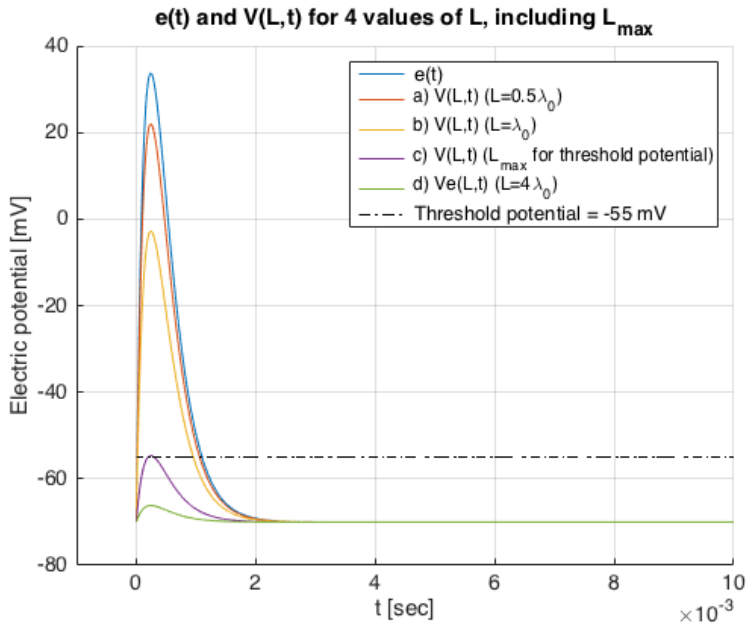


Figure 3.11: The input signal and the exact response $V(L,t)$ for 4 line lengths, around the length constant $\lambda_0 = \frac{1}{\sqrt{rg}} = 223$ μm, including the computed $L_{\max}$, for the model with ion pumps [125]. The dotted line represents the threshold potential of -55 mV.

Due to its dynamic nature, the neural signal has a greater attenuation than this because the RC line behaves as a low-pass filter, which attenuates furthermore the higher frequencies. The maximum transmission length $L_{\max}$ (the second characteristic parameter) can be computed approximately as the length of the line for which the standard neural signal is attenuated until its maximum value reaches the threshold potential: $V_t = -55$ mV:

$$F(0) = \frac{1}{\operatorname{ch}\left(\frac{L}{\lambda_0}\right)} = \frac{V_t - V_0}{V_{\max} - V_0}; \lambda_0 = \frac{1}{\sqrt{rg}} = \sqrt{\frac{ab}{2\sigma\rho}};$$

$$L_{max} = \lambda_0 \operatorname{arg\,cosh}\left(\frac{V_{max} - V_0}{V_t - V_0}\right) = \sqrt{\frac{ab}{2\sigma\rho}} \operatorname{argcosh}\left(\frac{V_{max} - V_0}{V_t - V_0}\right).$$

In this case study, the value of $L_{\max}$ computed with this approximate analytical relation is:

$$L_{max} = \lambda_0 \operatorname{argch}\left(\frac{V_{max} - V_{min}}{V_t - V_{\min}}\right) = 0.223 \cdot \operatorname{argcosh}\left(\frac{100}{15}\right) = 0.5764 \text{ mm}, \tag{3.21}$$

6% lower than the numerical value previously computed.

It should be noted that $L_{\max}$ is in agreement with the Schwann cell length reported in literature, of approximately 100 μm, more precisely between 20 μm and 400 μm [88], [143], but the value obtained is less than the value of 2 mm used in several simulations [144], [145], [146].

The same effect can be obtained by simulating the model without ion pumps, with an excitation of the form $e(t) + V_0$. In this case, the threshold potential will be $V_t = 15$ mV, and the amplitude that the signal must reach to reach the threshold potential is also 15

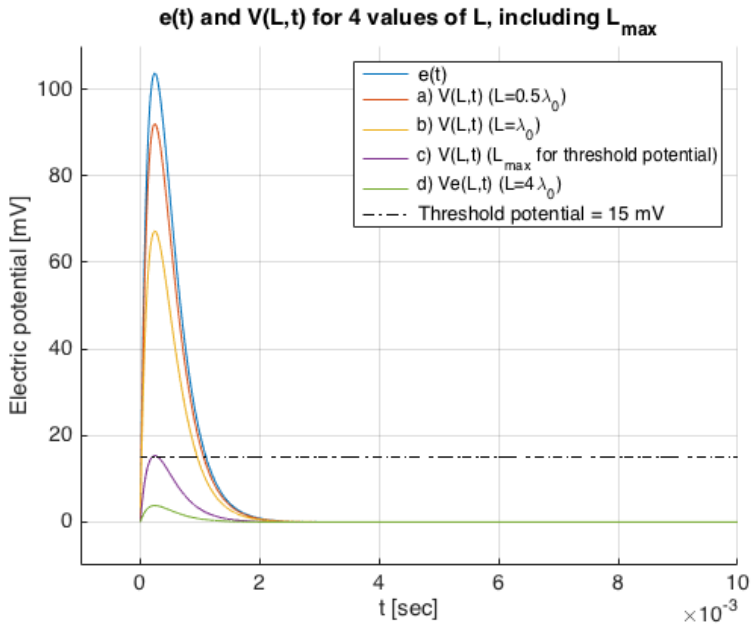


Figure 3.12: The input signal and the exact response $V(L,t)$ for 4 line lengths, around the length constant $\lambda_0 = \frac{1}{\sqrt{rg}} = 223$ μm, including the computed $L_{\max}$, for the model without ion pumps. Input: $e(t) + V_0$. The dotted line represents the threshold potential of 15 mV.

mV (15 mV – 0 mV = 15 mV). The length value corresponding to threshold is the same in this case as for the ion pumps model. Fig. 3.12 represents the time response for four line lengths, including $L_{\max}$. The absolute and the relative errors for a line length shorter than the length constant $L = 0.25\lambda_0 < \lambda_0$ are:

$$\text{err}_{\text{abs}} = 0.64;$$
$$\text{err}_{\text{rel}} = 0.006.$$

3.3.6 Characteristic parameters: signal delay and transmission speed

Numerical approach

Note that the time needed for the signal at the end of the line to reach the threshold potential depends on the line length L. Therefore the delay time can be defined as the amount of time needed for the signal to reach the threshold, assuming that the input signal reaches the threshold instantaneously (Fig. 3.13).

Fig. 3.14 represents the signal delay (defined as the moment the signal reaches the threshold), for different line lengths between $0.5\lambda_0$ and $L_{\max}$. The signal delay was computed using (3.20) for 6 line lengths (including $L_{\max}$) and the values were interpolated for 100 points; for line lengths beyond $L_{\max}$ the delay cannot be computed, as these signals do not reach the threshold at all. The derivative of the inverse of this function is actually the local transmission speed of the signal. We find that the local transmission speed decreases as the line length grows (Fig. 3.15). For $L = L_{\max}$, the transmission speed (computed as $\frac{\mathrm{d}L}{\mathrm{d}(t_d)}$ (using regressive first order finite differences) has the value of 0.43 m/s (0.37 m/s

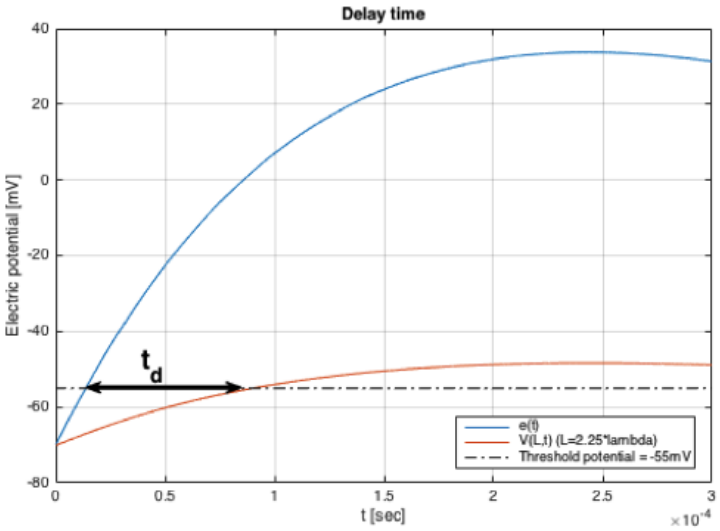


Figure 3.13: Definition of delay time, as the time needed for the signal to reach the threshold.

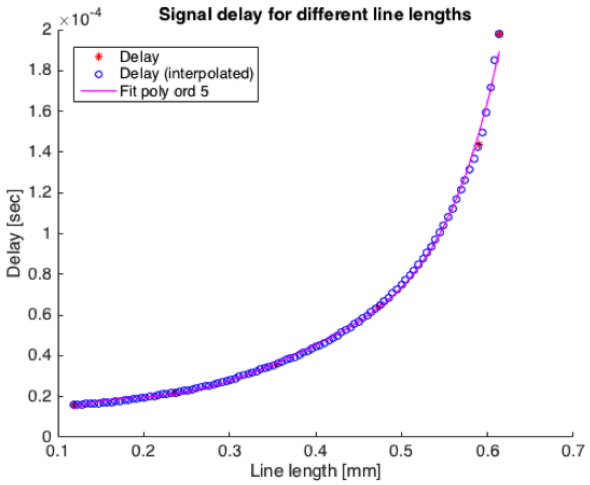


Figure 3.14: The signal's delay vs. the line length (between $0.5\lambda_0$ and $L_{\max}$, using (3.20).

using interpolated values), and for $L = L_0$, the transmission speed is around 19.66 m/s (52.86 m/s using interpolation). These values are in agreement with the ones reported in literature: [147], where for neurons having the axon's diameter of 5-8 μm the speed is between 4 and 24 m/s and for diameters of 13-20 μm, the speed is between 80 and 120 m/s.

An additional way to compute the delay and the transmission speed is by using (3.19), so as $\frac{\mathrm{d}x}{\mathrm{d}(t_d)}$, with L fixed. Fig. 3.16 and Fig. 3.17 present the delay and the transmission speed for 5 line lengths between $0.5\lambda_0$ and $L_{\max}$, computed using (3.19), with $L = L_{\max}$ and $x \in [0.5\lambda_0, L_{\max}]$. From Fig. 3.16 it can be deducted that by using this method the slope of the function delay vs. line length is higher in the vicinity of $0.5\lambda_0$, so the derivative of the inverse, the transmission speed respectively, will be smaller. Conversely, near $L_{\max}$ the situation is the opposite, which makes that the values of the transmission speed along the line to be closer to each other. For $L = L_{\max}$, the transmission speed is about 1.91 m/s (1.76 m/s using interpolated values) and for $L = L_0$ the transmission speed is around 6.55 m/s (9.58 m/s using interpolation).

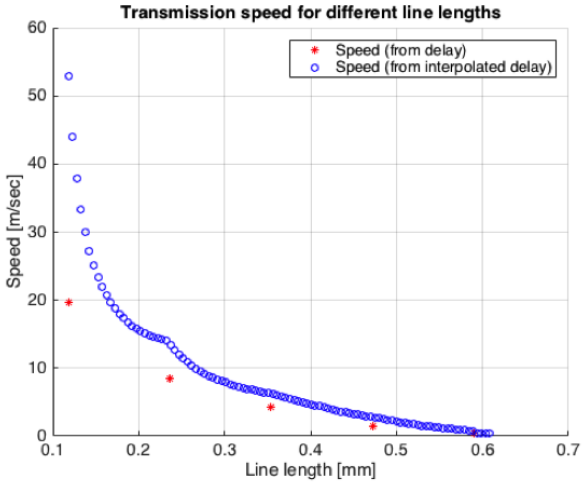


Figure 3.15: The local transmission speed vs. the line length (between $0.5\lambda_0$ and $L_{\max}$, using (3.20).

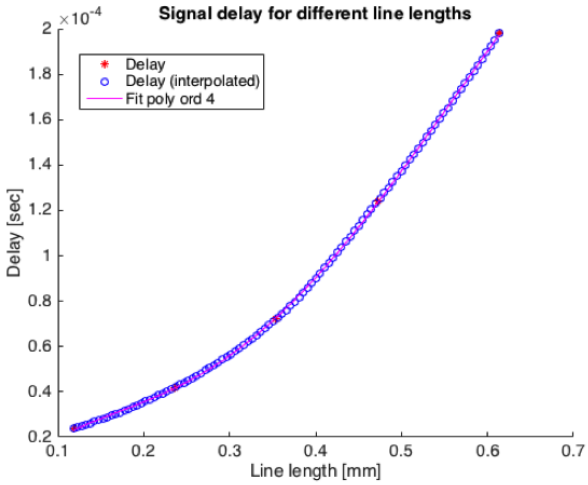


Figure 3.16: The signal's delay vs. the line length (between $0.5\lambda_0$ and $L_{\max}$, using (3.19).

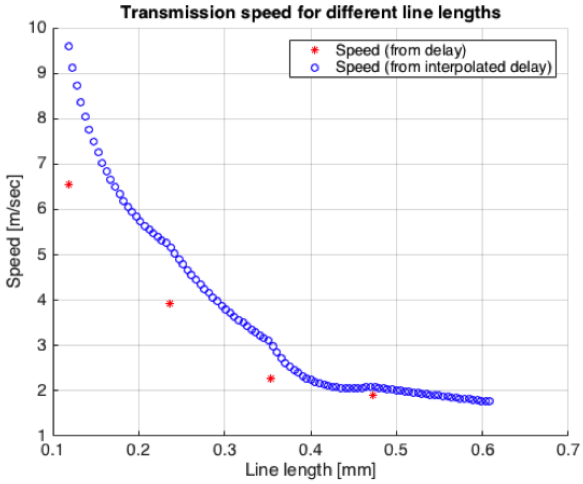


Figure 3.17: The local transmission speed vs. the line length (between $0.5\lambda_0$ and $L_{\max}$, using (3.19).

Analytical approach

The transmission speed can be approximated analytically by estimating t_d – the time with which the impulse is delayed by the RC line. The time t_d can be expressed by replacing the RC line with its Π equivalent scheme, consisting of the longitudinal resistance $R = rL$ in parallel with a capacitor $C/2 = cL/2$ at one end and a conductance $G/2 = gL/2$ at the other end placed transversally. In this simplified model, the transfer function of the line has the approximate expression (reduced model of order 1):

$$\mathrm{F}_a(s) = \frac{2}{(G+sC)(R+2/(G+sC))} = \frac{1}{sRC/2 + RG/2 + 1} = \frac{2/(RC)}{s + 1/\tau_0}. \tag{3.22}$$

- **Model 1.1 – approximation with rectangular impulse**

  If we approximate the input signal with a rectangular impulse of amplitude V_{max}, then the output signal is rising monotonously over time during the impulse:

$$E(s) = \frac{\mathrm{V}_{\max}}{s} \Rightarrow V(L,s) = \mathrm{F}_a(s)\mathrm{E}(s) = \frac{\mathrm{V}_{\max}}{s(sRC/2 + RG/2 + 1)} = \frac{2V_{\max}}{RC}\frac{1}{s\,(s+s_0)},$$

  with

$$s_0 = \frac{RG+2}{RC}; \frac{1}{s\,(s+s_0)} = \frac{A}{s} + \frac{B}{s+s_0} = \frac{A\,(s+s_0) + Bs}{s\,(s+s_0)};$$
$$A + B = 0, As_0 = 1 \Rightarrow A = -B = 1/s_0.$$

  Then:

$$V(L,t) = \mathcal{L}^{-1}(V(L,s)) = \frac{2V_{\max}}{s_0 RC}\mathcal{L}^{-1}\left(\frac{1}{s} - \frac{1}{s+s_0}\right) = \frac{2V_{\max}}{RG+2}\left(1 - e^{-s_0 t}\right).$$

  This expression for the output signal allows us to compute the delay time of the impulse as:

$$V\,(L,t_d) = V_t - V_0 \Longrightarrow 1 - e^{-s_0 t_d} = \frac{(V_t - V_0)\,(RG+2)}{2V_{\max}}$$
$$\Longrightarrow e^{-s_0 t_d} = 1 - \frac{(V_t - V_0)\,(RG+2)}{2V_{\max}}$$
$$\Rightarrow t_d = -\frac{\ln\left(1 - \frac{(V_t - V_0)(RG+2)}{2V_{max}}\right)}{s_0} = -\frac{\ln\left(1 - \frac{(V_t - V_0)(RG+2)}{2V_{max}}\right) RC}{RG+2}.$$

  The delay time is small in comparison with the rising time:

$$\begin{aligned} V(L,t) = \frac{2V_{\max}}{RG+2}\left(1 - e^{-s_0 t}\right) \approx \frac{2V_{\max} s_0 t}{RG+2} = V_t - V_0 \\ \Longrightarrow t_d \approx \frac{(V_t - V_0)\,RC}{2V_{\max}} = \frac{(V_t - V_0)\,rcL^2}{2V_{\max}}. \end{aligned} \tag{3.23}$$

  With this value for the delay, the transmission speed has the expression:

$$v_t = \frac{L}{t_d} = \frac{2V_{\max}}{(V_t - V_0)\,rcL} = \frac{V_{\max} ab}{(V_t - V_0)\,\rho L\varepsilon}, \tag{3.24}$$

where the line capacity and resistance have the values in Table 3.3.

In particular, for a line of 0.1 mm, the speed has the value of:

$$v_t = \frac{2V_{max}}{(V_t - V_0)\, rcL} = \frac{200}{15 \cdot 6 \cdot 10^9 \cdot 2 \cdot 10^{-9} \cdot 10^{-4}} = \frac{10^6}{90} = 10^4 \text{ m/s}.$$

In this model the transmission speed increases proportionally to the diameter of the axon and inversely proportional to the line length. The value obtained is much higher than the values reported in the literature, which shows that the model is not accurate. The explanation lies in the fact that in this model the rising time of the excitation signal is considered null.

- **Model 1.2 – approximation with triangular impulse**

  It is expected to get better results if we approximate the excitation signal with a triangular impulse of magnitude V_{max} and rising time τ. Now the excitation signal is the integral of the signal in Model 1.1:

$$\begin{aligned}
e_1(t) = \frac{V_{max}}{\tau} t &= \frac{1}{\tau} \int_0^t e\,(t')\, dt' \\
&\Longrightarrow \mathrm{E}_1(s) = \frac{\mathrm{E}(s)}{s\tau} \\
&\Longrightarrow V_1(L,s) = \frac{V(t,L)}{s\tau} \\
&\Longrightarrow V_1(L,t) = \frac{1}{\tau} \int_0^t V\,(t',L)\, dt' = \frac{2V_{\max}}{\tau(RG+2)} \int_0^t \left(1 - e^{-s_0 t}\right) dt' \\
&\Longrightarrow V_1(L,t) = \frac{2V_{\max}}{\tau(RG+2)} \left(t + \frac{1}{s_0}\left(e^{-s_0 t} - 1\right)\right) \approx \frac{V_{\max} s_0 t^2}{\tau(RG+2)}.
\end{aligned}$$

$$\begin{aligned}
V_1\,(L, t_d) = V_t - V_0 &\Rightarrow \frac{V_{max} s_0 t_d^2}{\tau(RG+2)} = V_t - V_0 \\
&\Longrightarrow t_d = \sqrt{\frac{\tau RC\,(V_t - V_0)}{V_{\max}}} = L\sqrt{\frac{\tau rc\,(V_t - V_0)}{V_{\max}}} = L\sqrt{\frac{2\tau\rho\varepsilon\,(V_t - V_0)}{V_{\max} ab}}.
\end{aligned}$$

In this model the transmission speed (which we call "the speed of thought" :-)) has the expression:

$$v_t = \frac{L}{t_d} = \sqrt{\frac{V_{max} ab}{2\tau\rho\varepsilon\,(V_t - V_0)}}. \tag{3.25}$$

For the test case studied, which has an outer diameter of $2(a+b) = 20$ microns, the speed has the value:

$$v_t = \frac{L}{t_d} = \sqrt{\frac{V_{max} ab}{2\tau\rho\varepsilon\,(V_t - V_0)}} = \sqrt{\frac{100 \cdot 7 \cdot 3 \cdot 10^{-12}}{2 \cdot 10^{-3} \cdot 0.92 \cdot 136 \cdot 10^{-12} \cdot 15}} = 23 \text{ m/s},$$

independent of the line length and proportional with the square root of the axon's diameter [125]. This kind of growth is also met in the specialized literature [148], [149].

Another aspect emphasized by (3.25) is the dependence between the speed and membrane thickness b, and in the case of myelinated axons the thickness of the myelin layer of the glial cell, respectively. The speed increase is proportional to the square root of b.

The thickness of the unmyelinated axon's membrane b is 1000 times smaller than the Schwann cells' thickness, which explains why the velocity in the saltatory conduction is about 30 times greater than in the unmyelinated axons. In the case of myelinated axons, the thickness b is in correlation with the diameter D ($D = 2(a+b) \approx 5b$), so **the velocity will be proportional to the outer diameter**, this correlation also being reported in the literature: *"The ratio of the inner (axon) perimeter to the outer (myelin) perimeter remains constant at or near the optimal value of 0.6 for conduction in all groups of fibres"* [150]. Timotin considers in [72] a ratio close to $d/D = 0.7$.

The relation (3.25) underlines the speed's sensitivity to the geometric and material parameters: the speed does not depend on the length of the fiber, increases proportionally with the diameter of the myelinated segment (for which the thickness of the myelin layer is proportional to the diameter of the cytoplasm) and decreases proportionally with the cytoplasmic resistivity's square root and with the permeability of the myelin layer. These results are in full agreement with those obtained numerically in [92]. The relative sensitivity (defined as the ratio between percentage change of velocity and percentage change of the parameter) is null for the fiber length (L), 1 for the outer diameter (D), $-1/2$ for the cytoplasmic resistivity (ρ) and for the permeability of the myelin layer (ε).

Thus, in a neuron with an outer diameter of 80 microns the transmission speed becomes 100 m/s. An increase in velocity can be obtained by considering other factors besides the thickness, such as the duration τ of the neural signal rising front or the material constants, especially the resistivity ρ of the neuron's membrane.

A different approach that allows the estimation of **the minimum transmission speed** uses the following approximate expression, valid for axons of maximal length:

$$v_t = \frac{L_{max}}{t_m + t_d} = \frac{\lambda_0}{t_m + t_d} \operatorname{arg\,cosh}\left(\frac{V_{\max} - V_{\min}}{V_t - V_{\min}}\right) \approx \frac{0.576}{0.243} = 2.37 \text{ m/s},$$

where

$$e'(t_m) = 0 \Rightarrow \frac{1}{\tau_1} e^{-\frac{t_m}{\tau_1}} = \frac{1}{\tau_2} e^{-\frac{t_m}{\tau_2}} \Rightarrow e^{t_m\left(\frac{1}{\tau_2} - \frac{1}{\tau_1}\right)} = \frac{\tau_1}{\tau_2}$$

$$\Longrightarrow t_m = \frac{\ln\left(\frac{\tau_1}{\tau_2}\right)}{\frac{1}{\tau_2} - \frac{1}{\tau_1}} = \frac{10^{-4} \ln\left(\frac{3}{2}\right)}{\frac{1}{2} - \frac{1}{3}} = 0.243 \text{ ms}.$$

The quantity t_m represents the time the signal needs to reach its peak (maximum amplitude). As expected, the speed has lower values in this model. If the fiber length is less than the maximum value, then the speed is lower because the threshold value is reached before the signal peak. Then we have:

$$v_t = \frac{L_{max}}{t_m + t_d} = \frac{1}{t_m + t_d} \sqrt{\frac{ab}{2\sigma\rho'}} \operatorname{argcosh}\left(\frac{V_{max} - V_0}{V_t - V_0}\right). \tag{3.26}$$

Table 3.4: The geometrical values for unmyelinated and myelinated axons.

	a – inner radius	b – thickness of membrane / Schwann cell
Unmyelinated axons	$a = D/2 - b \approx D/2$	$b = 3$ nm
Myelinated axons	$a = 2b/2$	$b = 0.2D$

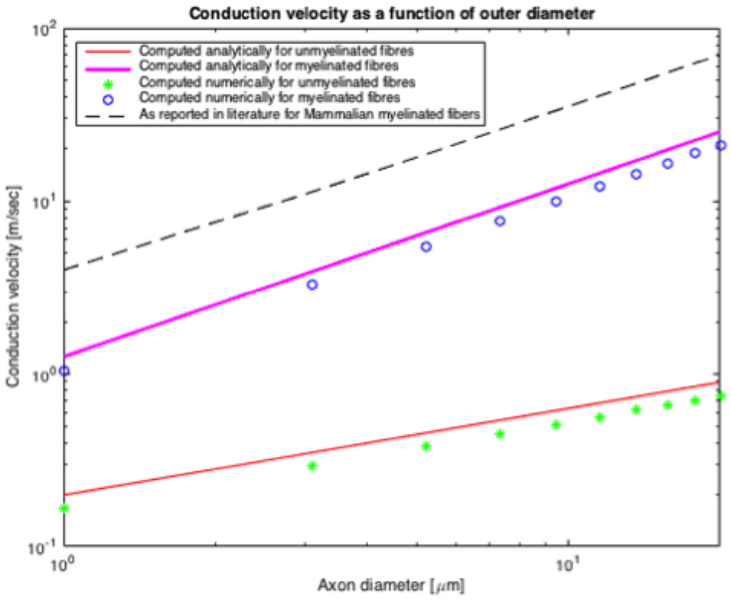


Figure 3.18: The conduction speed as a function of axon's diameter; the dotted black line is the dependence for Mammalian myelinated reproduced from [151].

In this model the speed also grows proportionally with the axon's diameter.

A more careful analysis of the delay time may find its value smaller than the time constant τ_0 and therefore a higher speed. References [148], [149] and [151] report the transmission speed values vs. fiber diameter for different species. For diameters around 7 microns the speeds are between 2 m/s and 30 m/s.

Local Transmission Speed vs. axon's outer diameter – numerically and analytically

We have determined numerically the dependence between the local conduction speed and axon's diameter by calculating the velocity for different outer diameters. The calculation of the speed was based on the delay defined as in Fig. 3.13, by varying the line length between $0.5\lambda_0 \div 3\lambda_0$ from the variation over time for:

$$V(L,s) = \frac{E(s)}{\mathrm{ch}(\lambda L)} + \frac{V_0(1 - 1/\mathrm{ch}(\lambda L))}{s}.$$

The numerical values used to determine the conduction speed are presented in Table 3.4.

The values obtained for the transmission parameters are shown in Table 3.5. It is interesting to remark that although the characteristic length λ_0 changes with diameter

Table 3.5: The transmission parameters computed numerically for axons of different diameters, for myelinated and unmyelinated axons.

	$D[\mu\text{m}]$	1	3.1	5.2	7.3	9.4	11.6	13.7	15.8	17.9	20
Unmyelinated	λ_0 [mm]	0.002	0.004	0.005	0.005	0.006	0.007	0.007	0.008	0.008	0.009
	$L_{\max}$ [mm]	0.005	0.009	0.012	0.014	0.016	0.018	0.019	0.021	0.022	0.023
	$\frac{L_{\max}}{\lambda_0}$ [-]	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
	vit_{λ_0} [m/s]	0.166	0.293	0.380	0.450	0.511	0.565	0.615	0.660	0.703	0.743
	$\text{vit}_{L_{\max}}$ [m/s]	0.033	0.059	0.077	0.09	0.1	0.113	0.123	0.132	0.141	0.15
Myelinated	λ_0 [mm]	0.013	0.039	0.066	0.093	0.119	0.146	0.172	0.199	0.226	0.252
	$L_{\max}$ [mm]	0.033	0.102	0.171	0.241	0.31	0.379	0.448	0.518	0.587	0.656
	$\frac{L_{\max}}{\lambda_0}$ [-]	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
	vit_{λ_0} [m/s]	1.051	3.271	5.490	7.710	9.929	12.148	14.368	16.587	18.807	21.026
	$\text{vit}_{L_{\max}}$ [m/s]	0.21	0.654	1.10	1.54	1.99	2.43	2.87	3.32	3.76	4.21

(this is normal because the parameters depend on a and b), the ratio between λ_0 and $L_{\max}$ remains constant. Fig. 3.18 shows the transmission speed vs. the axon's diameter for myelinated axons and unmyelinated axons. The dependency was determined both analytically and numerically, using the numerical values in Table 3.4. The analytical values were determined using (3.25).

The values computed for different diameters are close to the values reported in the literature for myelinated axons ([151], Figure 3, Mammalian myelinated); Fig. 3.18 is missing the dependence for unmyelinated axons reported in literature, due to the fact that the reported values are very diverse. It is found that the numerical values are close to those determined analytically; for myelinated axons, the values obtained (both analytically and numerically) are smaller than those reported in the literature, but the slope of growth (double logarithmic scale) is the same.

3.3.7 Simulation of the transmission line in LTspice

LTspice [152] provides a component called "Lossy Transission Line" (symbol: LTLIN) for the simulation of transmission lines, documented in [153]. This component allows the definition of the line parameters (resistance, capacity, inductivity, conductance) and the simulation of the transmission line (TL). We defined as input a voltage source variable over time by an exponential expression, so the form of the signal would be similar to the one resulting from (3.9), but with null initial condition $V_0 = 0$. Considering the length unit of 1 μm, the values of the line parameters (per unit length) are as follows: $c = 2$ fF/μm, $r = 6$ kΩ/μm, $g = 3$ nS/μm. In order to simulate a line length of about $4\lambda_0 = 892$ μm the number of length units (the parameter Len) needed is 1000.

We found that LTSpice does not allow the definition of a capacitor and a conductance at the same time (error: "Nonzero G (except RG) line not supported yet" and the RG parameter is not recognized).

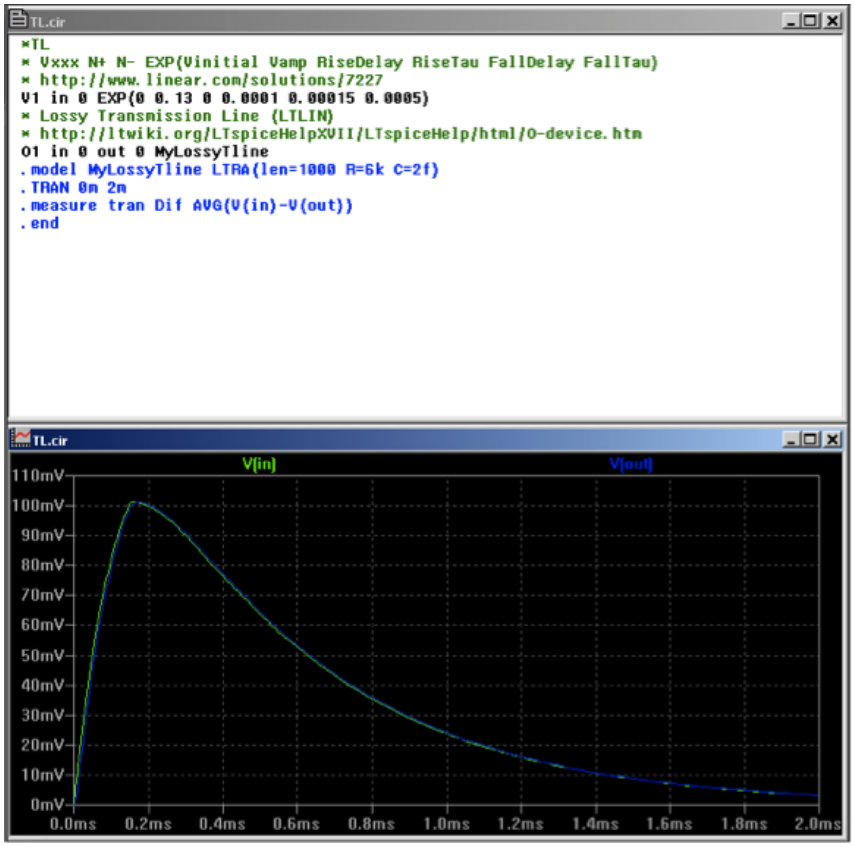


Figure 3.19: The simulation of the transmission line in LTspice – 1 mm.

Fig. 3.19 shows the netlist file and the results of the TL (with RC parameters) simulation for the time domain [0–2] ms.

As shown in Fig. 3.19, the amplitude of the output signal is very close to that of the input signal for the simulated line length, although according to previous calculations the output should be below the threshold of 15 mV. This may be due to the missing conductance in the LTspice model. Fig. 3.20 shows the simulation results for Len = 10000, which corresponds to a line length of 10 mm. The amplitude decreases to half and the delay is significantly higher compared to previous case.

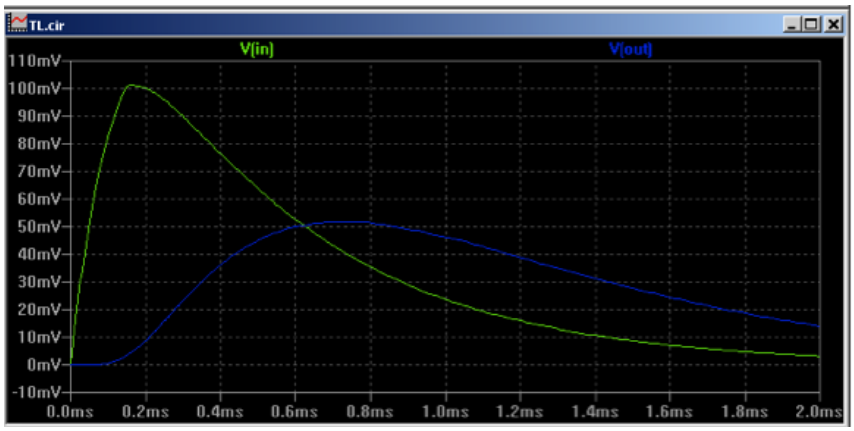


Figure 3.20: The simulation of the transmission line in LTspice – 10 mm.

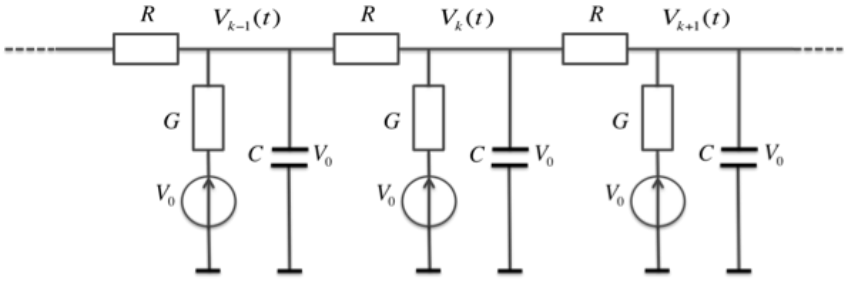


Figure 3.21: The companion circuit generated by the spatial discretization with centered differences.

3.4 Numerical 1D model

Apparently, the numerical solution is not necessary, since the problem allows finding an analytical solution. As the analytical solving can only be done in linear cases, the numerical study will be useful in nonlinear problem solving, such as those encountered in simulating the saltatory conduction in myelinated axons, in which Ranvier nodes introduce strong nonlinearities.

3.4.1 Discretization of the line equation with finite differences

By discretizing the second order derivative from (3.18) using finite centered differences, we get:

$$\frac{V_{k-1}(t) - 2V_k(t) + V_{k+1}(t)}{\Delta x^2} = \frac{1}{\lambda_0^2}\left(V_k(t) - V_0\right) + \frac{\tau}{\lambda_0^2}\frac{\mathrm{d}V_k(t)}{\mathrm{d}t}$$
$$\Longrightarrow V_{k-1}(t) - 2V_k(t) + V_{k+1}(t) = \frac{\Delta x^2}{\lambda_0^2}\left(V_k(t) - V_0\right) + \Delta x^2\frac{\tau}{\lambda_0^2}\frac{\mathrm{d}V_k(t)}{\mathrm{d}t}.$$

With $\frac{\Delta x^2}{\lambda_0^2} = rg\Delta x^2 = RG, \Delta x^2\frac{\tau}{\lambda_0^2} = rc\Delta x^2 = RC$, where $R = r\Delta x, G = g\Delta x, C = c\Delta x$, it follows that:

$$V_{k-1}(t) - 2V_k(t) + V_{k+1}(t) = RG\left(V_k(t) - V_0\right) + RC\frac{\mathrm{d}V_k(t)}{\mathrm{d}t}$$

$$\Longrightarrow -\frac{1}{R}V_{k-1}(t) + \left(\frac{2}{R} + G + sC\right)V_k(t) - \frac{1}{R}V_{k+1}(t) = GV_0 \tag{3.27}$$

The expression (3.27) represents the nodal analysis equation for the circuit in Fig. 3.21.

Consequently, the numerical model generated is a reduced model having the order equal to the number of internal nodes of the spatial grid.

This model is not an optimal one, so the numerical resolution should be done with control of the numerical error while keeping the computational effort to a minimum. Order reduction through optimal segmentation is based on intelligent meshing with error control. This is done by estimating the dependence between the truncation error and the spatial step (which gives the order of the reduced system).

3.4.2 Discretization error

The dependence between the error and the refinement of the spatial mesh can be estimated analytically.

From the Taylor expansion of the solution in the grid nodes having the step $h = \Delta x$ we obtain the relation [154]:

$$u_{i+1} = u_i + u_i^{(1)}h + \frac{1}{2}u_i^{(2)}h^2 + \frac{1}{6}u_i^{(3)}h^3 + \frac{1}{24}u_i^{(4)}h^4 + O\left(h^5\right),$$
$$u_{i-1} = u_i - u_i^{(1)}h + \frac{1}{2}u_i^{(2)}h^2 - \frac{1}{6}u_i^{(3)}h^3 + \frac{1}{24}u_i^{(4)}h^4 + O\left(h^5\right).$$

By summation, this gives:

$$u_{i+1} + u_{i-1} = 2u_i + u_i^{(2)}h^2 + \frac{1}{12}u_i^{(4)}h^4 + O\left(h^5\right)$$
$$\Longrightarrow u_i^{(2)} = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{h^2}{12}u_i^{(4)} + O\left(h^3\right).$$

By naming V_k and U_k the numerical and the analytical solutions in the node k, quantities that in stationary regime satisfy $\lambda_0 \frac{V_{k-1}-2V_k+V_{k+1}}{h^2} = V_k$, respectively $\lambda_0^2 \left.\frac{\partial^2 U}{\partial x^2}\right|_{x=x_k} = U_k$, it follows that the deviation between them has the expression:

$$E_k = V_k - U_k = \lambda_0^2 \left(\frac{V_{k-1} - 2V_k + V_{k+1}}{h^2} - \left.\frac{\partial^2 U}{\partial x^2}\right|_{x=x_k} \right) = \lambda_0^2 \frac{h^2}{12} u_k^{(4)} + O\left(h^3\right)$$
$$\Longrightarrow E_k = V_k - U_k = \lambda_0^2 \frac{h^2}{12} u^{(4)}\left(\xi\right)\big|_{\xi\in(x_k, x_{k+1})}$$

and consequently the numerical error is bounded by:

$$|E_k| = |V_k - U_k| \leq \frac{\left(\lambda_0 h\right)^2}{12} c_4 = kh^2, \text{ where } c_4 = \max_{0<x<L}\left|\frac{\partial^4 V}{\partial x^4}\right|, \text{ so } k = \frac{\lambda_0^2 c_4}{12}.$$

As previously shown, the stationary solution has for unitary potential input the expression:

$$V(x) = -\frac{1}{\mathrm{sh}(\lambda L)}\left(-\frac{1}{\mathrm{ch}(\lambda L)}\mathrm{sh}(\lambda x) + \mathrm{sh}(\lambda(x-L))\right), \text{ with } \lambda = 1/\lambda_0.$$

By derivation, we determine c_4:

$$\frac{\partial^4 V}{\partial x^4} = \frac{\lambda^4}{\mathrm{sh}(\lambda L)}\left(\frac{1}{\mathrm{ch}(\lambda L)}\mathrm{sh}(\lambda x) - \mathrm{sh}(\lambda(x-L))\right) \Rightarrow c_4 = 1/\lambda_0^4.$$

In conclusion, the error bound can be estimated as:

$$|E_k| = |V_k - U_k| \leq \frac{\left(h/\lambda_0\right)^2}{12} = \frac{\left(L/\left(\lambda_0(N-1)\right)\right)^2}{12} = \frac{\left(L/\lambda_0\right)^2}{12(N-1)^2} = \frac{0.6188}{(N-1)^2}. \quad (3.28)$$

For 50 nodes, the error should have the order $2.6 \cdot 10^{-4}$ and for 10 nodes it should be less than 1%.

This result can also be used to estimate the error of reduced order models obtained by segmentation. Consequently, the 10-segment models have acceptable accuracy for most practical applications. In fact, this is the recommendation made by several papers on numerical simulation for the number of compartments for myelinated section [92].

Since the fourth derivative is proportional to the solution (with the ratio of $1/\lambda_0^4$), we expect that the local error to be maximal at the beginning of the line, and at the end to have values up to 15% of the maximum value. The expression of the error margin has a local nature and is valid for any node, so the inequality is also satisfied by the Cebisev norm of the error:

$$\|E\|_\infty = \max_{k=1,N} |V_k - U_k| \leq \frac{(h/\lambda_0)^2}{12} c_4 = kh^2.$$

A similar relationship is satisfied by the Euclidean norm (more precisely by the root mean square deviation):

$$\|E\|_2 = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(V_k - U_k)^2} \leq \sqrt{\frac{1}{N}\sum_{k=1}^{N}(kh^2)^2} \leq \sqrt{(kh^2)^2} \leq kh^2 = \frac{(\lambda_0 h)^2}{12}\rfloor_4.$$

If we take into account the approximation:

$$\begin{aligned}\frac{\partial^4 V}{\partial x^4} &= \frac{\lambda^4}{\mathrm{sh}(\lambda L)}\left(\frac{1}{\mathrm{ch}(\lambda L)}\mathrm{sh}(\lambda x) - \mathrm{sh}(\lambda(x-L))\right) \approx \\ &\approx \frac{\lambda^4}{\exp(\lambda L)}\left(\frac{2(\exp(\lambda x) - \exp(-\lambda x))}{\exp(\lambda L)} - \exp(\lambda(x-L)) + \exp(\lambda(L-x))\right) = \\ &= \lambda^4((\exp(\lambda x) - 2\exp(-\lambda x))\exp(-2\lambda L) + \exp(-\lambda x)) \approx \lambda^4 \exp(-\lambda x),\end{aligned}$$

we obtain a smaller bound for the error:

$$\begin{aligned}\|E\|_2 &= \sqrt{\frac{1}{N}\sum_{k=1}^{N}(V_k - U_k)^2} \leq \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\frac{(\lambda_0 h)^2}{12}c_4\right)^2} \leq \frac{(\lambda_0 h)^2}{12}\sqrt{\frac{1}{L}\int_0^L c_4^2(x)\mathrm{d}x} = \\ &= \frac{(h/\lambda_0)^2}{12}\sqrt{\frac{1}{L}\int_0^L \exp\left(-2x/\lambda_0\right)\mathrm{d}x} = \frac{(h/\lambda_0)^2}{12}\sqrt{\frac{\lambda_0}{2L}\left(1 - \exp\left(-2L/\lambda_0\right)\right)} = \frac{(h/\lambda_0)^2}{28},\end{aligned} \tag{3.29}$$

which is at least two times smaller than previously estimated by (3.28).

The truncation error depends on the spatial discretization step through the relation [155]:

$$\mathrm{TE}_{\mathrm{rel}} = k_x \cdot \Delta x^2, \text{ with } k_x \text{ fixed .}$$

In order to determine the constant k_x we solved the electric potential equation in stationary regime with null initial conditions, considering a step input E=1 mV.

3.4.3 Verification of numerical model accuracy

In stationary regime with null initial conditions, (3.17) becomes:

$$\frac{\partial^2 V(x,t)}{\partial x^2} = rgV(x,t). \tag{3.30}$$

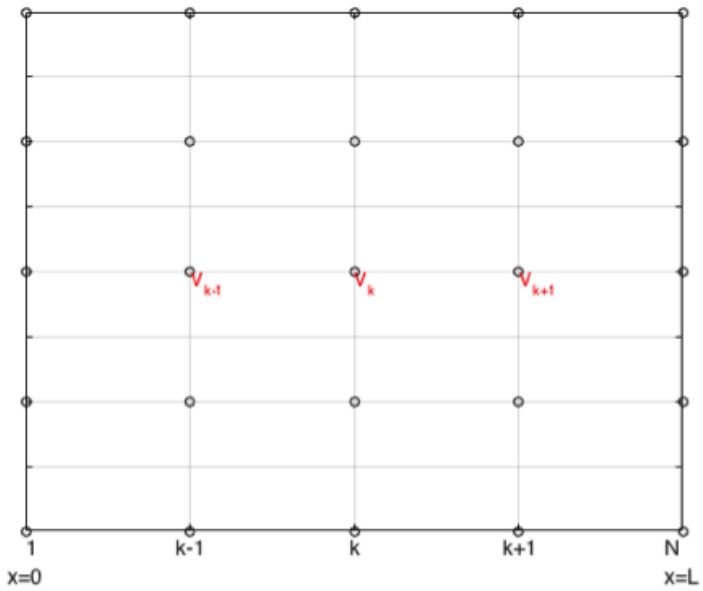


Figure 3.22: Discretization with constant step using centered finite differences, network 1.

Equation (3.30) is discretized using centered finite differences as:

$$\frac{2}{(x_{k-1}-x_k)(x_{k-1}-x_{k+1})}V_{k-1}+\frac{2}{(x_k-x_{k-1})(x_k-x_{k+1})}V_k+ \\ +\frac{2}{(x_{k+1}-x_{k-1})(x_{k+1}-x_k)}V_{k+1}=rgV_k.$$

We defined two dual networks as follows:

– Network 1: N nodes, constant step Δx, as in Fig. 3.22.

The equation (3.30) becomes:

$$\frac{V_{k-1}-2V_k+V_{k+1}}{(\Delta x)^2}=rgV_k.$$

The boundary condition at $x = L$ is Neumann and is determined from the first derivative as follows:

$f(\mathrm{x}) = ax^2 + bx + c$

with $f(-h) = ah^2 - bh + c = f_1; f(0) = c = f_2; f(h) = ah^2 + bh + c = f_3$

$\Longrightarrow ah^2 - bh = f_1 - f_2; ah^2 + bh = f_3 - f_2$

$\Longrightarrow a = (f_1 - 2f_2 + f_3) / \left(2h^2\right); b = (-f_1 + f_3)/(2h)$

$\Longrightarrow f'(\mathrm{x}) = 2ax + b$

with $f'(\mathrm{h}) = 2ah + b = (f_1 - 2f_2 + f_3 - f_1/2 + f_3/2)/h = (f_1/2 - 2f_2 + 3f_3/2)/h.$

This expression for the boundary condition provides an error of order $O(h^2)$ which is similar to that provided by the other equations. If regressive differences were used, in which only the last two nodes intervene, then the error would have the order $O(h)$ and the solution of the whole linear system would be compromised.

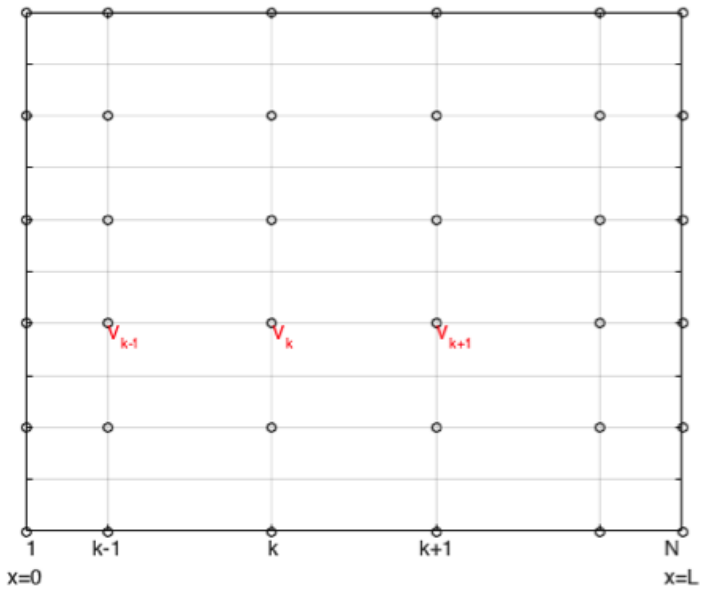


Figure 3.23: Discretization with network 2.

The system to be solved has the unknowns $V_k, k = 1 : N$, the free term is null (with the exception of the first position = 1 mV) and the coefficients matrix is tridiagonal $\mathbf{A}_1 \in \mathbb{R}^{N\times N}$ of the following form (example for $N = 5$):

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} \\ 0 & 0 & 1/2 & -2 & 3/2 \end{bmatrix}$$

– Network 2: the nodes are at the middle between the nodes of network 1, therefore we have $N + 1$ nodes, a constant step Δx between the nodes 2 and N; between nodes $1 \to 2$ and $N \to N + 1$, the step will be $\Delta x/2$ (Fig. 3.23).

This time the coefficients matrix $\mathbf{A}_2 \in \mathbb{R}^{(N+1)\times(N+1)}$ will have the following form (example for $N = 5$):

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{2}{3(\Delta x/2)^2} & -\frac{1}{(\Delta x/2)^2} - rg & \frac{1}{3(\Delta x/2)^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{3(\Delta x/2)^2} & -\frac{1}{(\Delta x/2)^2} - rg & \frac{2}{3(\Delta x/2)^2} \\ 0 & 0 & 1/2 & -2 & 3/2 \end{bmatrix}$$

Table 3.6 represents the stationary potential along the line for the dual networks and the distribution of the stationary potential along the line, computed analytically with the following expression:

$$V(x) = -\frac{1}{\mathrm{sh}(2L)}\left(-\frac{1}{\mathrm{ch}(\lambda L)}\,\mathrm{sh}(\lambda x) + \mathrm{sh}(\lambda(x-L))\right), \text{ with } \lambda = 1/\lambda_0.$$

Table 3.6: The stationary potential along the line for dual spatial networks.

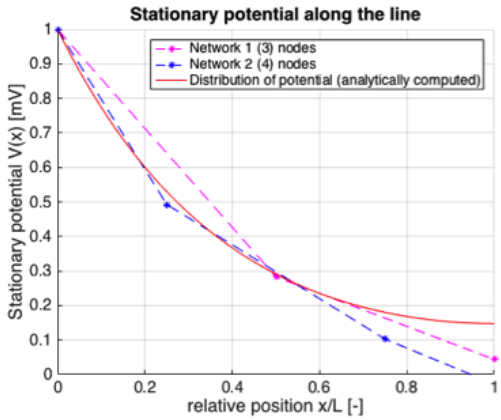
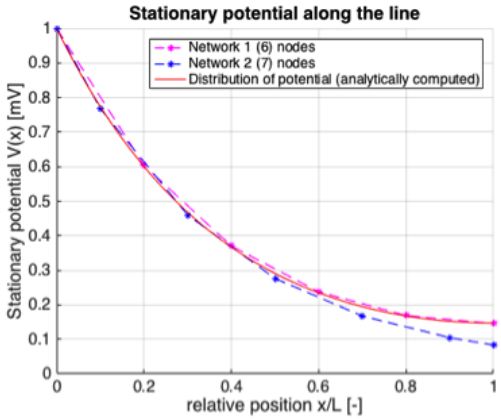
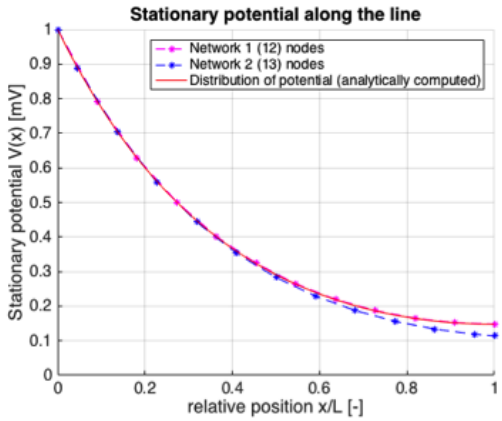
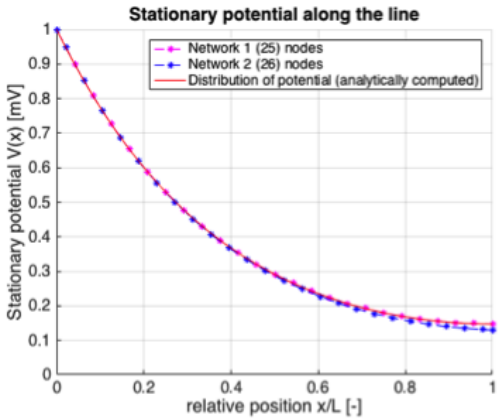
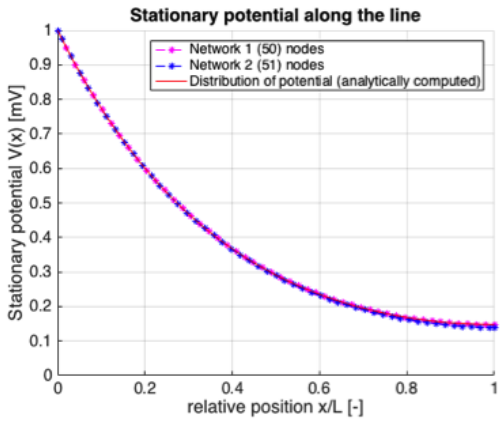
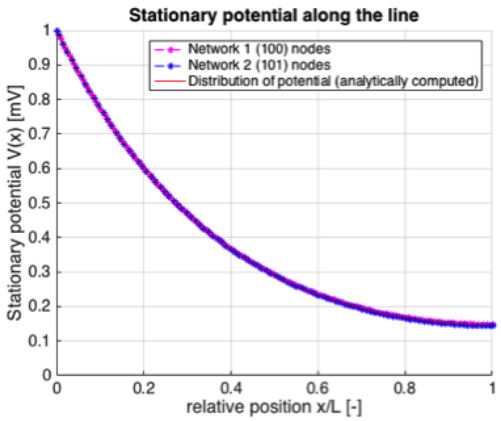
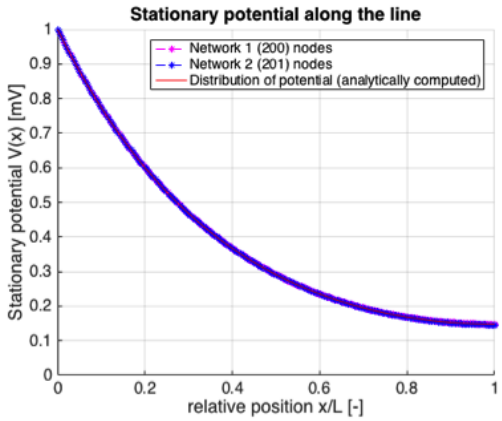
3 nodes 	6 nodes 
12 nodes 	25 nodes 
50 nodes 	100 nodes 
200 nodes 	

Table 3.7: The relative deviations between the potential at the end of the line ($x = L$), numerical and exact.

no. of nodes	3	6	12	25	50	100	200
bound of relative error	0.17129	0.06312	0.03435	0.01757	0.00911	0.00465	0.00235
deviation from exact value	0.10388	0.00147	0.00039	0.00014	0.00004	0.00001	0.000003

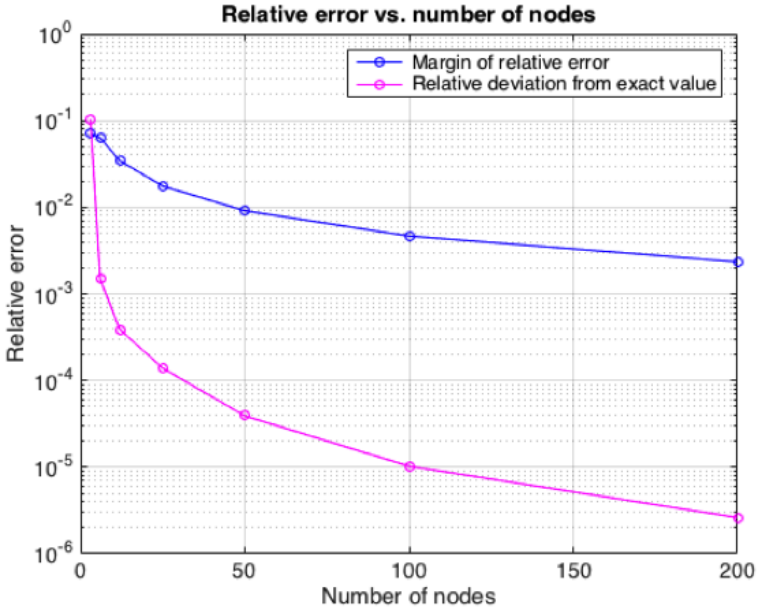


Figure 3.24: The relative error and its upper bound as a functions of the number of nodes of the spatial network.

The number of nodes increases in geometric progression: 3, 6, 12, 25, 50, 100, 200.

Table 3.7 contains the relative deviation between the potential at the end of the line ($x = L$) computed numerically with the primary and the dual network, and the deviation between the primary network and the analytically computed value, with the expressions:

$$\text{bound of relative error} = \frac{|V_{x=L,\ \text{net1}} - V_{x=L,\text{net2}}|}{V_{x=0,\ \text{net1}}}; \tag{3.31}$$

$$\text{deviation from exact value} = \frac{|V_{x=L,\ \text{net1}} - V_{x=L,\text{analyt}}|}{V_{x=0,\ \text{analyt}}}. \tag{3.32}$$

Fig. 3.24 shows the dependence of the relative error margin in Table 3.7 vs. the number of nodes in the spatial network. Note that the margin of the relative difference between the potentials obtained using the dual networks is less than 10% at 6 nodes and at most 1% starting from 50 nodes. In reality the deviation between the numerical simulation and the exact result is much smaller.

For each spatial network we have checked if the values at the middle and the end of the line of the analytically determined stationary potential line are framed by the values

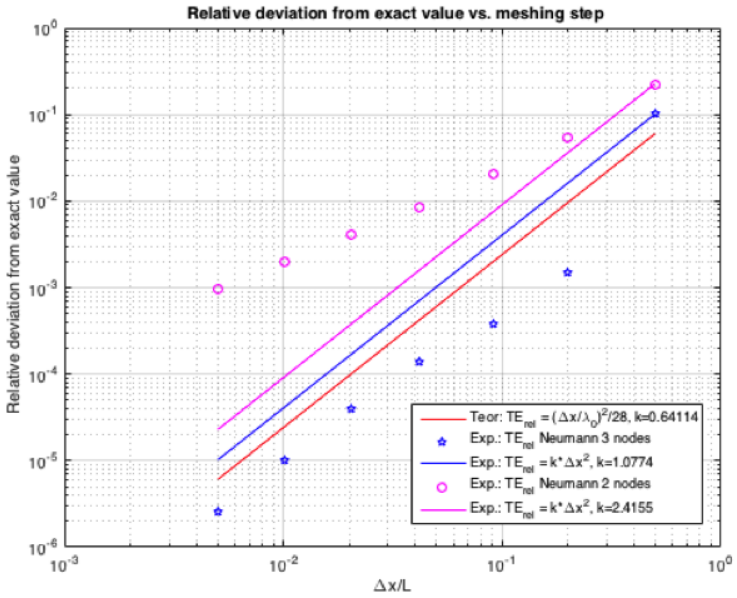


Figure 3.25: The relative error vs. the refinement of the spatial network (experimentally and theoretically).

determined by the two networks (for even numbers of nodes we computed the middle value by linear interpolation of the neighboring nodes). For 3 and 6 nodes, the analytical values at the end of the line are not framed, but for all other cases, the analytical value is framed between network 1 (higher value) and network 2 (lower value). The absolute errors at the end of the line between network 1 and analytical are very small, even at 12 nodes, as illustrated by Fig. 3.24.

Based on the dependence between the relative error bound and the number of internal nodes of the spatial mesh, we can determine the minimal network that provides an acceptable error. With a 6-nodes mesh, an acceptable error of about 0.1% is obtained.

If the Neumann boundary condition for $x = L$ would have been discretized with regressive finite differences of first order, then just the last two nodes would be used in the BC, therefore the last line of the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ would be:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Since in this case the error of the last equation has the order $O(h)$ (unlike the other equations in the system for which the error has the order $O(h^2)$), then the slope of the relative error margin is approximately 2, and not 1 as theoretically approximated.

Fig. 3.25 shows the dependence of the relative error on the spatial network step, theoretically (red) – calculated with (3.29) – and experimentally – calculated as the deviation between the exact and numerical solutions (for Neumann BC at $x = L$ discretized with regressive finite differences of order 1 (magenta) and with regressive finite differences of order 2 (blue)).

Fig. 3.25 points out that the numerical method has the order of convergence two, if the Neumann BC is discretized with the same order as that of the internal nodes and that the whole solution has the order one, when the Neumann condition is discretized with

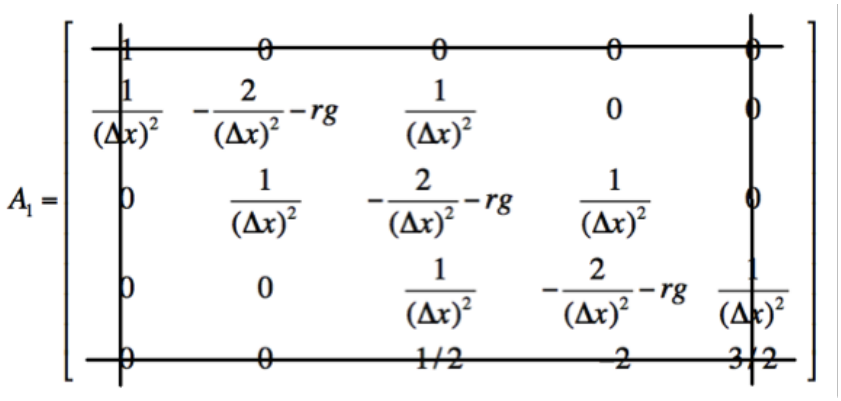


Figure 3.26: Obtaining $\mathbf{A}_\mathrm{n}$ from $\mathbf{A}_1$.

regressive finite differences of the first order, thus the order of the BC monopolizes the entire solution.

Another aspect that deserves to be analyzed is that concerning the characteristics of the matrix system, both in the case of $\mathbf{A}_1$ and $\mathbf{A}_2$. These are tridiagonal matrices, none of them being symmetric, even if the nodal matrix of the electric circuit in Fig. 3.21 has these two remarkable properties.

The nodal matrix ($\mathbf{A}_\mathrm{n}$) can be extracted from $\mathbf{A}_1$ by eliminating the first and last column and line (Fig. 3.26), in which case the free term of the system would have on the first position the short-circuit current $e(t)/R$; also, if we were to modify $\mathbf{A}_\mathrm{n}(N-1, N-1) = -1/(\Delta x)^2 - rg$, it would mean having a Neumann BC of order 1, so the numerical method would have the order of convergence 1 (the solution values would overlay the magenta markers in Fig. 3.25).

In order to determine a symmetrical matrix $\mathbf{A}_\mathrm{n}$ that preserves the equation coefficients of $\mathbf{A}_1$ we proceeded as follows:

- The first line of $\mathbf{A}_1$ represents the equation V_1 = excitation = 1 mV. Thus by eliminating the first line and column, the new first position of the free term would be $-\frac{1}{(\Delta x)^2}$.
- The last two lines encode the equations:

$$\frac{1}{(\Delta x)^2}V_{N-2} + \left(-\frac{2}{(\Delta x)^2} - rg\right)V_{N-1} + \frac{1}{(\Delta x)^2}V_N = 0;$$

$$V_{N-2} - 4V_{N-1} + 3V_N = 0 \Rightarrow V_N = \frac{(4V_{N-1} - V_{N-2})}{3}.$$

By eliminating V_N from the first equation, it follows that:

$$V_{N-2}\frac{2}{3(\Delta x)^2} + V_{N-1}\left(-\frac{2}{3(\Delta x)^2} - rg\right) = 0.$$

We wish that the coefficient of V_{N-2} to be $\frac{1}{(\Delta x)^2}$, for the matrix $\mathbf{A}_\mathrm{n}$ to remain symmetrical, we thus rewrite the previous equation as:

$$\frac{2}{3}\left(V_{N-2}\frac{1}{(\Delta x)^2} + V_{N-1}\left(-\frac{1}{(\Delta x)^2} - \frac{3rg}{2}\right)\right) = 0.$$

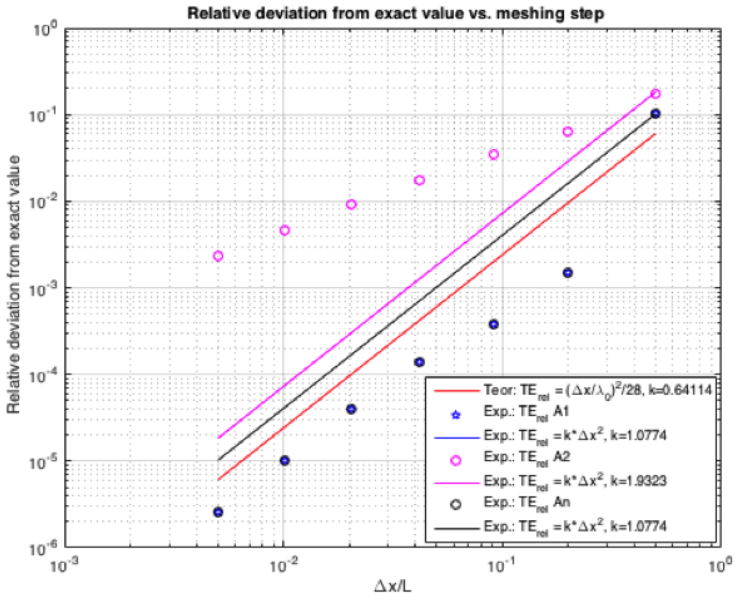


Figure 3.27: The relative error vs. the refinement of the spatial network for the solution obtained using system matrices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_\mathrm{n}$.

In conclusion, $\mathbf{A}_\mathrm{n} \in \mathbb{R}^{(N-2)\times(N-2)}$ will have the following form:

$$\mathbf{A}_\mathrm{n} = \begin{bmatrix} -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} & 0 & 0 & 0 \\ \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg & \frac{1}{\Delta x^2} \\ 0 & 0 & 0 & \frac{1}{\Delta x^2} & -\frac{1}{\Delta x^2} - \frac{3rg}{2} \end{bmatrix}$$

The free term will be $\mathbf{b}_\mathrm{n} \in \mathbb{R}^{(N-2)\times 1}$, having on the first position the value $-\frac{1}{\Delta x^2}$ and zero in rest. After determining the solution $\in \mathbb{R}^{(N-2)\times 1}$, we will complete it with the values V_1 and V_N from the eliminated equations.

In Fig. 3.27 the relative deviations from the exact values depending on the network refinement for the matrices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_\mathrm{n}$ are compared. The solution obtained with $\mathbf{A}_\mathrm{n}$ is identical to that obtained with $\mathbf{A}_1$. This was expected since the equations solved are the same. Therefore we can obtain the same solution by using a symmetrical tridiagonal system matrix.

3.4.4 Discretization of the standard neuronal signal

Up until this moment we considered an input signal approximated with the expression (3.9). It is possible that the transmission parameters previously determined depend on the form of the excitation signal. To check this hypothesis we chose from the literature four neural signals that we will consider as excitation signal. We will compare the transmission parameters computed by solving the equation (3.17) for these signals and for the excitation previously used, which will be considered as reference signal. The standard neural signals selected appear in the form of curves in figures, so they have to be digitized in order to be used in the numerical procedure.

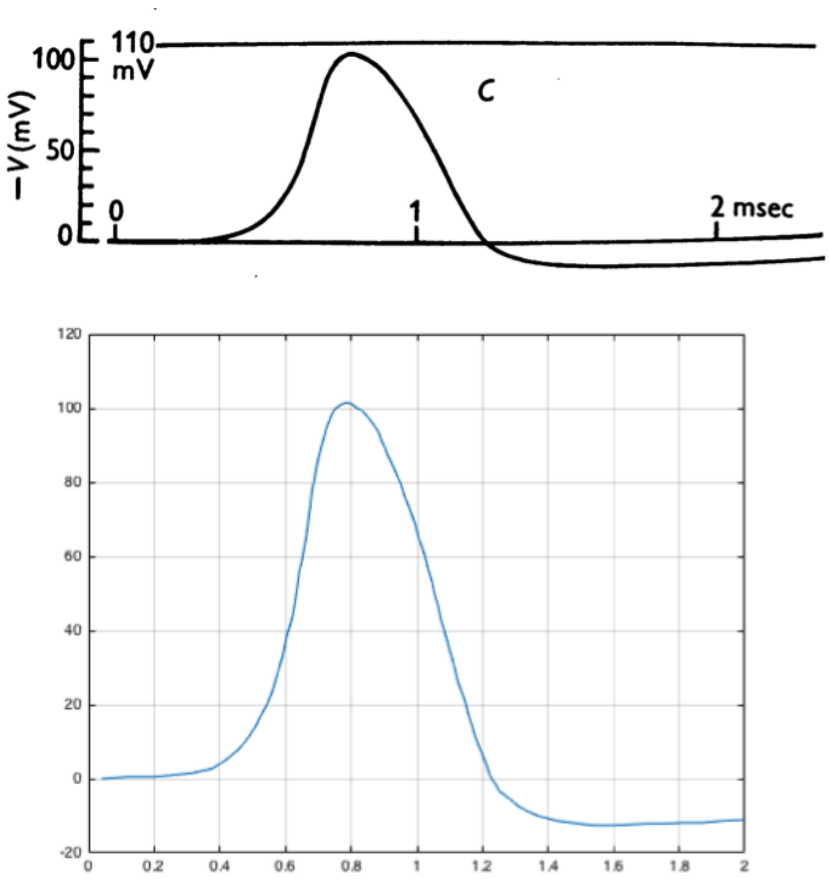


Figure 3.28: The theoretical signal represented in Fig. 15 from [98] and the signal digitized (100 points).

In most references just the scale is provided, but not V_0. In the first reference ([98]) the potential starts at 0 because the authors make a variable change; in the next two references only the reference unit is given, since the shape and the amplitude are more important. The signals were translated vertically to have the initial value $V_0 = -70$ mV as the reference signal.

For digitization we used the *digitizeGraph.m*, a semi-interactive tool that can be found at [156].

1. Digitizing Fig. 15 from [98].

   Two variations were reported in [98], one experimental and one theoretical. As the two signals are very similar, we use in what follows only the theoretical signal. Fig. 3.28 shows this signal and the digitized signal with 100 points.

2. Digitizing Figure 3C from [95].

   The original and digitized signals are shown in Fig. 3.29. The measure unit in this case is 50 mV (0y) and the total simulation time is 2 ms (0x).

3. Digitizing Figure 1B from [157].

   The original and digitized signals are shown in Fig. 3.30.

4. Digitizing Figure 2 from [96].

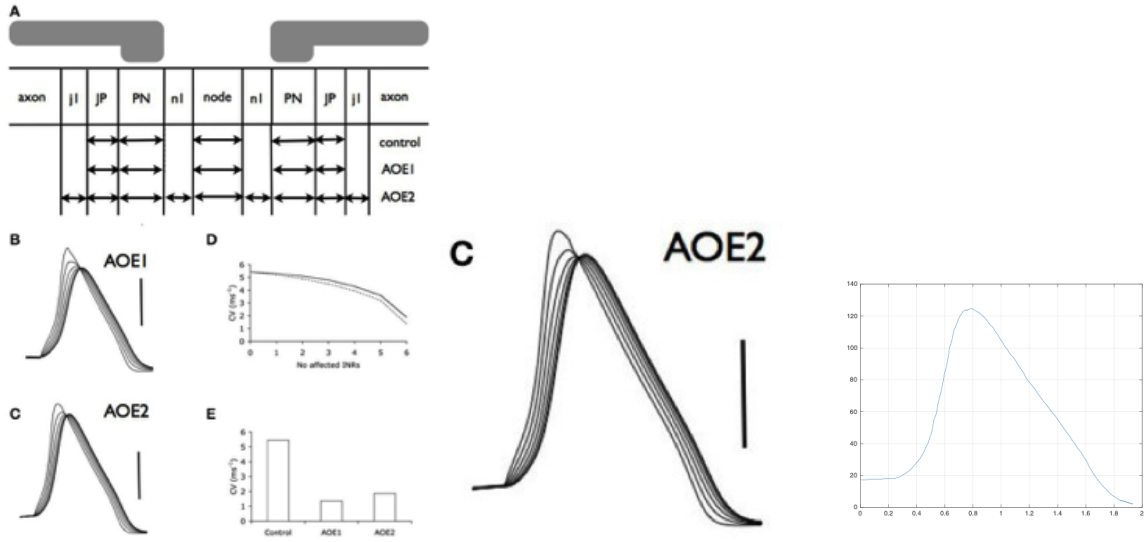


Figure 3.29: The original signal represented in Fig. 3C from [95] (right: zoom from C) and the signal digitized with 100 points.

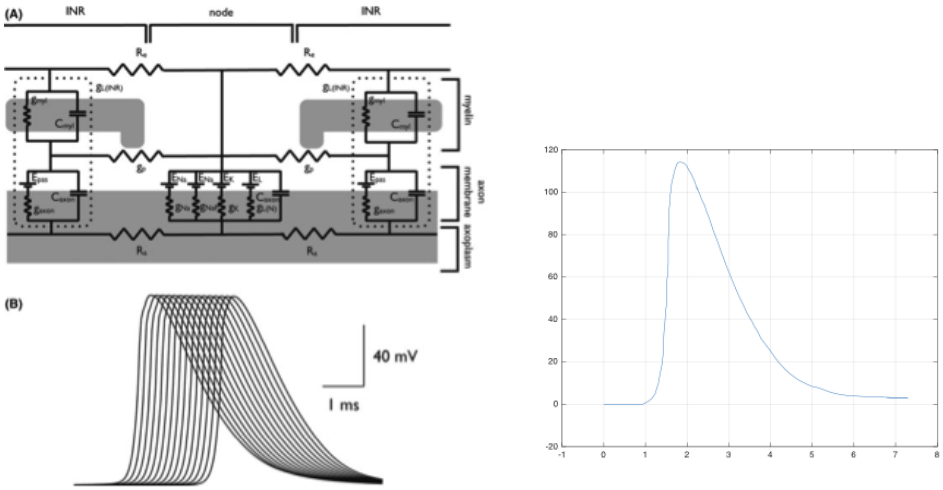


Figure 3.30: The original signal represented in Fig. 1B from [157] (right: zoom from C) and the signal digitized with 100 points.

The original and digitized signals are shown in Fig. 3.31.

The four signals digitized, before any processing, are represented in Fig. 3.32.

3.4.5 The results of the numerical simulation

We computed the transmission parameters (maximum transmission length, transmission speed) for 10 nodes, with two different excitations: the reference signal and one of the four signals obtained by discretization of the standard signal in the previous section.

The transmission parameters were calculated from the numerical solving of the equation (3.17), using the definitions in Section 3.3, with the difference that here, in the calculation of the derivative of the delay inverse to determine the transmission speed, we consider the origin (0, 0) as the first node. If two successive nodes were used it would mean $\mathrm{d}x(\mathrm{idx_x} - \mathrm{idx_x} + 1)/(\mathrm{d}t(\mathrm{idx_t} - \mathrm{idx_t} + 1))$, so the speed would always be equal to

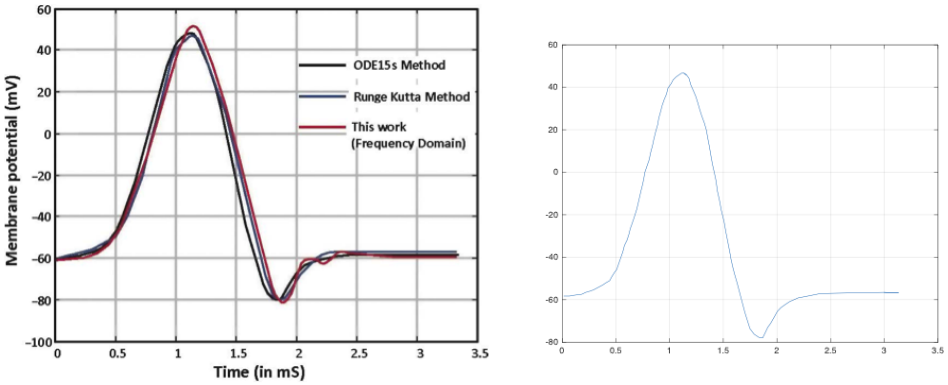


Figure 3.31: The original signal represented in Fig. 2 from [96] and the signal digitized with 100 points.

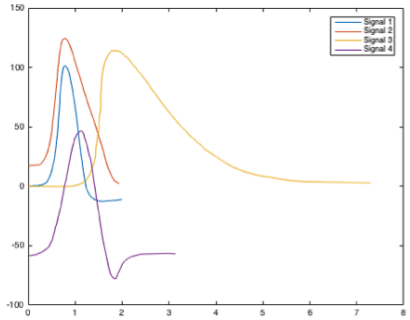


Figure 3.32: The four signals digitized with 100 points, with no post-processing.

$\mathrm{d}x/\mathrm{d}t$. For this reason, the derivative thus calculated is less precise than the previously calculated numerical value. However the purpose of this section is to compare the transmission parameters computed in the same way for different excitations, for this reason the precision of the computation is not very important.

For the excitation signal obtained by digitization to be consistent with the excitation signal used as reference, a correction of the discretized signals (vertical translation) is required by moving them down on the Oy axis, as follows: signals 1 and 3 with V_0, signal 3 with $V_0 - 17$ mV, signal 4 with -12 mV. The time domain of the 4 signals is considered 2 ms.

Table 3.8 contains the transmission parameters calculated numerically for pairs of excitations: the reference signal and the signal obtained by discretizing the standard signal. Because the speed is a local quantity, we give the results for the delay time around the characteristic length λ_0 and around the maximum length $L_{\max}$.

The maximum line length for which the signal can reach the threshold is the same for all signals, and in accordance with $L_{\max}$ computed analytically and with literature data.

The conduction speed for the reference signal around the characteristic length λ_0 is approximately 5 times smaller than the analytical velocity previously computed (23 m/s). However the comparison is limited since the speed computed here is a local quantity, which depends on the line length, whereas the analytical speed does not.

We remind the reader that in the computation of the delay time and transmission

Table 3.8: The transmission parameters numerically computed for different excitation signals.

Parameter / Excitation	$L_{\max}$	Amplitude	t_d at λ_0	Speed at λ_0	t_d at $L_{\max}$	Speed at $L_{\max}$
	[mm]	[mV]	[ms]	[m/s]	[ms]	[m/s]
Reference signal	0.5507	100	0.051	4.67	0.192	2.86
Digitized signal 1	0.5507	113.94	0.576	0.41	0.778	0.71
Digitized signal 2	0.5507	122.41	0.535	0.44	0.758	0.72
Digitized signal 3	0.5507	114.18	0.394	0.6	0.515	1.07
Digitized signal 4	0.5507	124.43	0.434	0.54	0.677	0.81

velocity in Section 3.3, we considered that the delay time of the standard signal is null because it was negligible. But here we deal with excitation signals that have delay times that cannot be neglected. To avoid a horizontal translation (to the left) of the digitized signals, we determined their delay time with respect to the delay time of the reference signal.

For each digitized signal an approximation obtained by exponential regression was extracted. The general model has the form $a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2}$. Table 3.9 shows the regression for each of the four digitized signals and the values for the coefficients of the general model (in the parenthesis the trust interval of 95%).

Table 3.10 shows the transmission parameters computed numerically for pairs of excitations: the reference signal and the signal obtained by discretizing the standard signal, this time by taking into account the delay time of the reference signal, thus computing the delay time of the response as the difference between the moment when it reaches the threshold and the time when the excitation signal reaches the threshold.

The conclusions concerning $L_{\max}$ and the transmission speed are similar to those drawn for Table 3.8. Compared to the local speed previously computed, these values lie between the value around $L_{\max}$ of 0.816 m/s and the value around λ_0 of 34.4 m/s. The difference may be due to the way the derivative was computed, less precise than if we considered closer nodes.

We find that the delay time and implicitly the transmission speed depend on the behavior of the input signal. The difference between the delay times of the output signals is explained by the delay time of the input signals (Table 3.11). The transmission speed is, as expected, inversely proportional to the delay time.

3.4.6 Numerical time integration

The equation (3.17) is parabolic, which means that unlike for hyperbolic equations (such as the wave equation), the explicit numerical integration methods are not very appropriate [154]. However, it is expected that the explicit and implicit methods (based on regressive differences, Crank-Nicolson) to have similar truncation errors for small enough time steps [155].

Table 3.9: The exponential regression for different excitation signals.

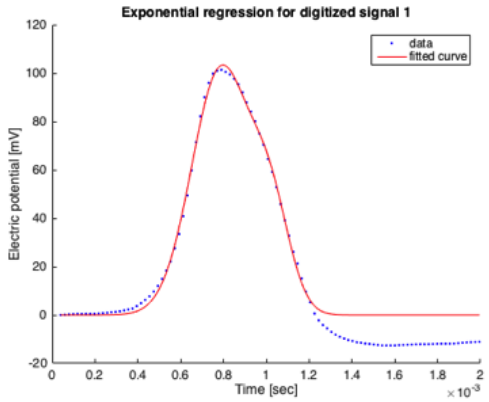
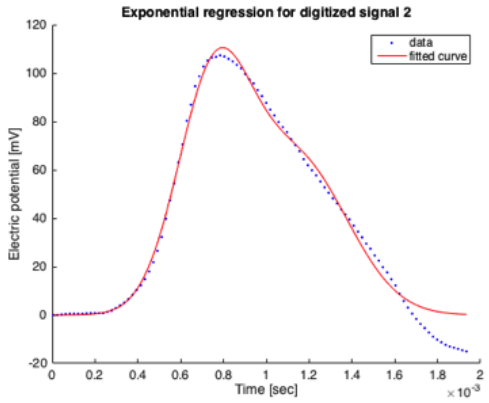
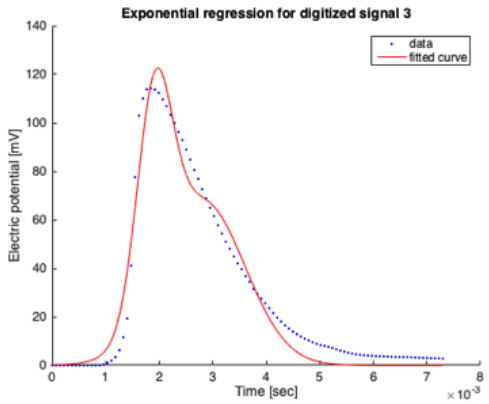
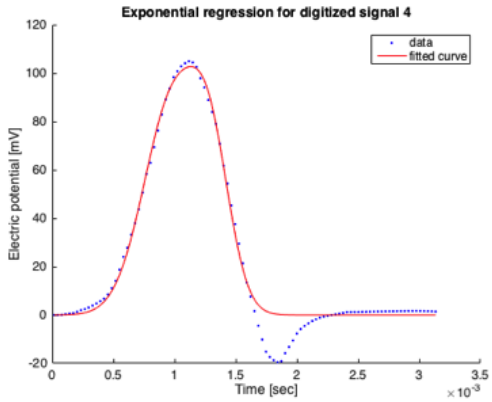
	$a_1 = 101.7 \quad (91.27, 112.1)$
	$b_1 = 0.0007879 \ (0.0007472, 0.0008286)$
	$c_1 = 0.0001925 \ (0.0001556, 0.0002295)$
	$a_2 = 38.63 \quad (12.02, 65.24)$
	$b_2 = 0.001021 \ (0.000966, 0.001076)$
	$c_2 = 0.000129 \ (7.704 \cdot 10^{-5}, 0.000181)$
	$a_1 = 91.87(72.65, 111.1)$
	$b_1 = 0.0007525(0.0007235, 0.0007815)$
	$c_1 = 0.0002411(0.0002174, 0.0002647)$
	$a_2 = 63.32 \quad (54.62, 72.02)$
	$b_2 = 0.001145(0.001065, 0.001225)$
	$c_2 = 0.0003364(0.0002722, 0.0004006)$
	$a_1 = 86.61 \quad (68.66, 104.6)$
	$b_1 = 0.001909(0.001878, 0.00194)$
	$c_1 = 0.0004611(0.0003824, 0.0005398)$
	$a_2 = 67.41 \quad (60.51, 74.31)$
	$b_2 = 0.002818(0.002622, 0.003014)$
	$c_2 = 0.001107(0.0009403, 0.001274)$
	$a_1 = 47.78(-18.36, 113.9)$
	$b_1 = 0.001303(0.001221, 0.001386)$
	$c_1 = 0.0002248(0.0001261, 0.0003234)$
	$a_2 = 89.65(58.47, 120.8)$
	$b_2 = 0.0009933(0.0008311, 0.001155)$
	$c_2 = 0.0003414(0.0002365, 0.0004464)$

Table 3.10: The transmission parameters numerically computed for different excitation signals, with taking into account the delay time of the excitation signal.

Parameter / Excitation	$L_{\max}$	Amplitude	t_d at λ_0	Speed at λ_0	t_d at $L_{\max}$	Speed at $L_{\max}$
	[mm]	[mV]	[ms]	[m/s]	[ms]	[m/s]
Reference signal	0.5507	100	0.01	23.36	0.152	3.63
Digitized signal 1	0.5507	113.94	0.051	4.67	0.253	2.18
Digitized signal 2	0.5507	122.41	0.071	3.33	0.293	1.87
Digitized signal 3	0.5507	114.18	0.030	7.78	0.152	3.63
Digitized signal 4	0.5507	124.43	0.071	3.33	0.313	1.75

Solving the equation (3.17) by discretization with three numerical time integration methods assumes meshing as follows:

$$\frac{V_{k-1}^{(j)} - 2V_k^{(j)} + V_{k+1}^{(j)}}{(\Delta x)^2} = rgV_k^{(j)} - rgV_0 + rc\frac{V_k^{(j+1)} - V_k^{(j)}}{\Delta t},$$

where k is the spatial index and j represents the temporal index.

a) **Backward-Time, Centered Space – BTCS** (implicit method, a system of equations is solved at every time step), the system of equations is given by:

$$\frac{V_{k-1}^{(j)} - 2V_k^{(j)} + V_{k+1}^{(j)}}{(\Delta x)^2} = rgV_k^{(j)} - rgV_0 + rc\frac{V_k^{(j)} - V_k^{(j-1)}}{\Delta t}.$$

The system matrix and the free term are as follows (example for size 5):

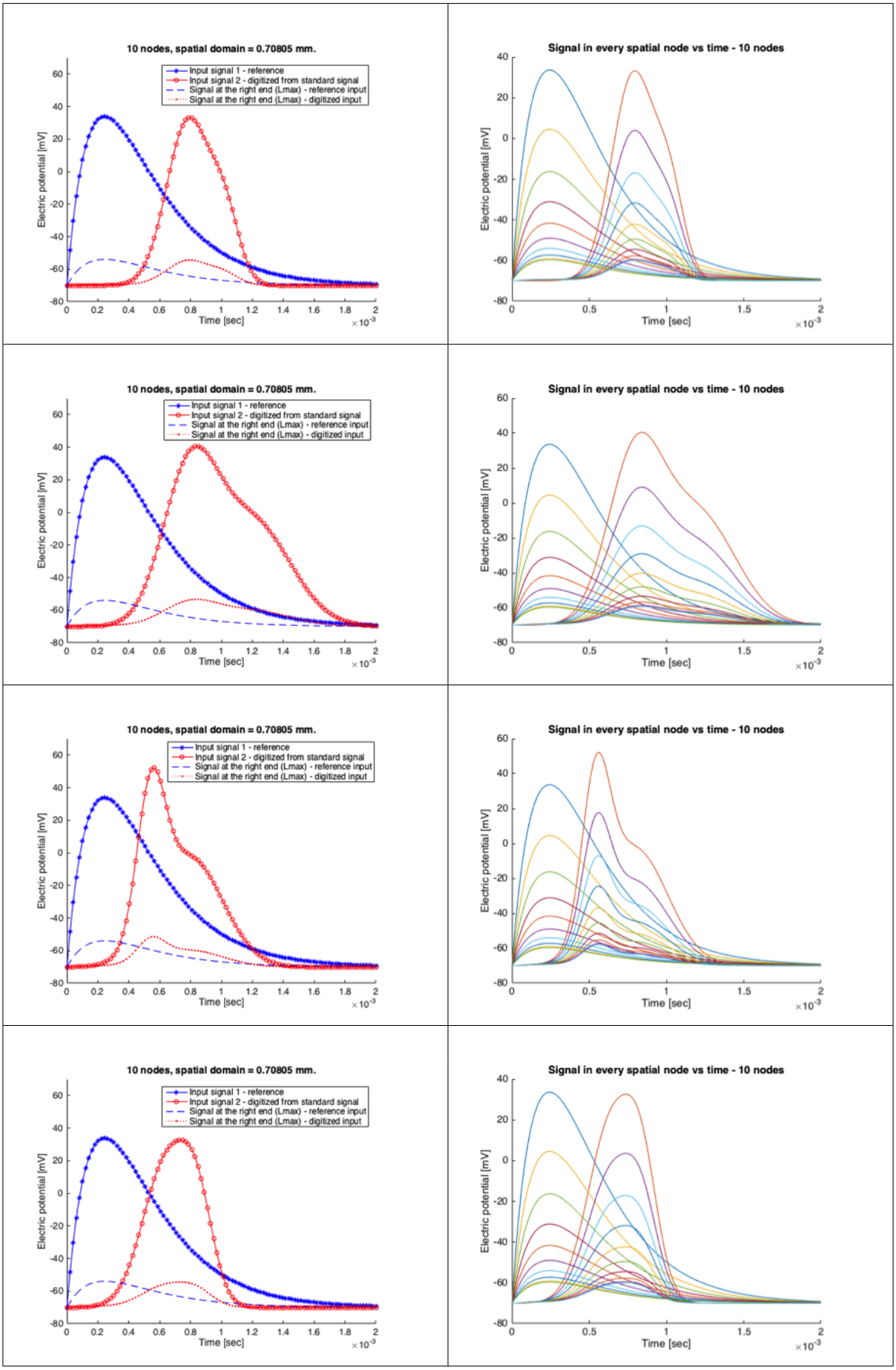
$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{\Delta x^2} \\ 0 & 0 & 1/2 & -2 & 3/2 \end{bmatrix};$$

$$\mathbf{TL}_1 = \begin{bmatrix} e(t) \\ .. \\ -\frac{rc}{\Delta t}V_k^{(j-1)} - rgV_0 \\ .. \\ 0 \end{bmatrix}.$$

From $\mathbf{A}_1$ we can extract the symmetrical matrix $\mathbf{A}_\mathrm{n}$ (the vector of free terms will be modified in consequence) which represents the matrix of coefficients for the same system of equations. Then, for $N = 5$, the size of matrix $\mathbf{A}_\mathrm{n}$ will be 3:

$$\mathbf{A}_\mathrm{n} = \begin{bmatrix} -\frac{2}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{\Delta x^2} & 0 \\ \frac{1}{\Delta x^2} & -\frac{2}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{\Delta x^2} \\ 0 & \frac{1}{\Delta x^2} & -\frac{1}{\Delta x^2} - \frac{3rg}{2} - \frac{3rc}{2\Delta t} \end{bmatrix};$$

Table 3.11: Numerical results for different excitation signals (input-output and reference-digitized).



$$\mathbf{TL}_{\mathrm{n}} = \begin{bmatrix} -\frac{1}{\Delta x^2}e(t) - rgV_0 - \frac{rc}{\Delta t}V_2^{(j-1)} \\ .. \\ \frac{rc}{\Delta t}V_k^{(j-1)} - rgV_0 \\ .. \\ -1.5\frac{rc}{\Delta t}V_N^{(j-1)} - 1 - 1.5rgV_0 \end{bmatrix}.$$

b) **Crank-Nicolson – CN** (implicit method, a system of equations is solved at every time step), the system of equations is given by:

$$\frac{1}{2}\left[\frac{V_{k-1}^{(j)} - 2V_k^{(j)} + V_{k+1}^{(j)}}{(\Delta x)^2} + \frac{V_{k-1}^{(j-1)} - 2V_k^{(j-1)} + V_{k+1}^{(j-1)}}{(\Delta x)^2}\right] = rgV_k^{(j)} - rgV_0 + rc\frac{V_k^{(j)} - V_k^{(j-1)}}{\Delta t}.$$

We deduce the system matrix and the free term as:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2(\Delta x)^2} & -\frac{1}{(\Delta x)^2} - rg - \frac{rc}{\Delta t} & \frac{1}{2(\Delta x)^2} & 0 & 0 \\ 0 & \frac{1}{2(\Delta x)^2} & -\frac{1}{(\Delta x)^2} - \frac{1}{(\Delta x)^2} - \frac{rc}{\Delta t} & \frac{1}{2(\Delta x)^2} & 0 \\ 0 & 0 & \frac{1}{2(\Delta x)^2} & -\frac{1}{(\Delta x)^2} - \frac{rc}{\Delta t} & \frac{1}{2(\Delta x)^2} \\ 0 & 0 & 1/2 & -2 & 3/2 \end{bmatrix};$$

$$\mathbf{TL}_1 = \begin{bmatrix} e(t) \\ .. \\ -\frac{1}{2(\Delta x)^2}V_{k-1}^{(j-1)} + \left(\frac{1}{(\Delta x)^2} - \frac{rc}{\Delta t}\right)V_k^{(j-1)} - \frac{1}{2(\Delta x)^2}V_{k+1}^{(j-1)} - rgV_0 \\ .. \\ 0 \end{bmatrix},$$

with the same notations as for BTCS.

The symmetrical matrix $\mathbf{A}_{\mathrm{n}}$ and the vector of free terms extracted are the following:

$$\mathbf{A}_{\mathrm{n}} = \begin{bmatrix} -\frac{1}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{2(\Delta x)^2} & 0 \\ \frac{1}{2(\Delta x)^2} & -\frac{1}{\Delta x^2} - rg - \frac{rc}{\Delta t} & \frac{1}{\Delta x^2} \\ 0 & \frac{1}{2(\Delta x)^2} & -\frac{1}{2(\Delta x)^2} - \frac{3rg}{2} - \frac{3rc}{2\Delta t} \end{bmatrix};$$

$$\mathbf{TL}_{\mathrm{n}} = \begin{bmatrix} -\frac{1}{2(\Delta x)^2}e(t) - \frac{1}{2(\Delta x)^2}V_1^{(j-1)} + \left(\frac{1}{(\Delta x)^2} - \frac{rc}{\Delta t}\right)V_2^{(j-1)} - \frac{1}{2(\Delta x)^2}V_3^{(j-1)} - rgV_0 \\ .. \\ -\frac{1}{2(\Delta x)^2}V_{k-1}^{(j-1)} + \left(\frac{1}{(\Delta x)^2} - \frac{rc}{\Delta t}\right)V_k^{(j-1)} - \frac{1}{2(\Delta x)^2}V_{k+1}^{(j-1)} - rgV_0 \\ .. \\ -\frac{3}{4(\Delta x)^2}V_{N-2}^{(j-1)} + \left(\frac{3}{2(\Delta x)^2} - \frac{3rc}{2\Delta t}\right)V_{N-1}^{(j-1)} - \frac{3}{4(\Delta x)^2}V_N^{(j-1)} - 1.5rgV_00 \end{bmatrix}.$$

c) **Forward-Time Centered Space – FTCS** (explicit method).

Table 3.12: Number of nodes and discretization steps.

Δx fixed	space		time		Δt fixed	space		time	
	N	Δx [m]	M	Δt [s]		N	Δx [m]	M	Δt [s]
	10	1.049e-4	112200	1.783e-8		5	2.360e-4	448800	4.456e-9
			448800	4.456e-9		10	1.049e-4		
			1009800	1.981e-9		15	6.743e-5		
			1795200	1.114e-9		20	4.969e-5		

The two implicit methods (BTCS and CN) are unconditionally stable, whereas, according to [155] and [158], FTCS is stable for:

$$r = \frac{\Delta t}{(\Delta x)^2} < \frac{1}{2}, \tag{3.33}$$

which means extremely small steps in the time domain $0, t_{\max} = [0, 2 \cdot 10^{-3}]$ sec and in the space domain $[0, 4\lambda_0]$. In addition, the truncation error for all three methods is dependent on the time and spatial discretization steps, as follows [155], [158]:

$$\text{err}_{\text{rel,FTCS}} = \mathcal{O}(\Delta t) + \mathcal{O}\left(\Delta x^2\right); \tag{3.34}$$

$$\text{err}_{\text{rel,BTCS}} = \mathcal{O}(\Delta t) + \mathcal{O}\left(\Delta x^2\right); \tag{3.35}$$

$$\text{err}_{\text{rel,CN}} = \mathcal{O}(\Delta t^2) + \mathcal{O}\left(\Delta x^2\right). \tag{3.36}$$

We compare the three methods based on the difference from the exact solution and on the solving time, with Δt fixed and Δx variable and vice versa. Using each of the three methods, we determine the transmission parameters: $L_{\max}$, the delay time, the transmission speed.

For the stability constraint to be satisfied for FTCS, we choose Δx, then compute $\Delta t = \frac{(\Delta x)^2}{2}$. The discretization steps used in tests are contained in Table 3.12.

The error was computed for the values corresponding to the last moment in time ([155]), as deviation from the exact solution, using the expression:

$$\text{err}_{\text{rel}} = \frac{\|V_e(:, t_{\max}) - V_n(:, t_{\max})\|}{\max(V_e(:, t_{\max}))} = \frac{\sqrt{\frac{\sum_k |V_e(k, t_{\max}) - V_n(k, t_{\max})|^2}{N}}}{\max(V_e(:, t_{\max}))},$$

with V_n – numerical value of the electric potential; N – number of spatial nodes.

The transmission parameters computed have values close to one another, $L_{\max} = 0.45$ mm, the delay time $t_d \in [1.36 \cdot 10^{-5}, 2 \cdot 10^{-4}]$ sec, the time increasing with the spatial variable (the lowest delay time is recorded for the smallest x, the highest for $L_{\max}$), the transmission speed $v_t \in [2, 6]$ m/s.

The figure 3.33 shows the relative error vs. the discretization step (Δx, Δt) for the three methods discussed and another numerical integration method from Matlab: *pdepe* [159]. It is found that the truncation errors for the four methods are almost identical for Δt fixed. For Δx fixed at $1.049 \cdot 10^{-4}$ and Δt between $1.11 \cdot 10^{-9}$ and $1.78 \cdot 10^{-8}$ (Fig. 3.33 left), the spatial truncation error has the order 10^{-4} and its contribution is significant in comparison to the temporal one, of order 10^{-6}, this is why according to (3.34-3.36) the three methods have comparable total truncation errors, with the order for Δt fixed between 10^{-4} and 10^{-2}.

In terms of computing time FTCS is relatively faster (Fig. 3.34). Because the dis-

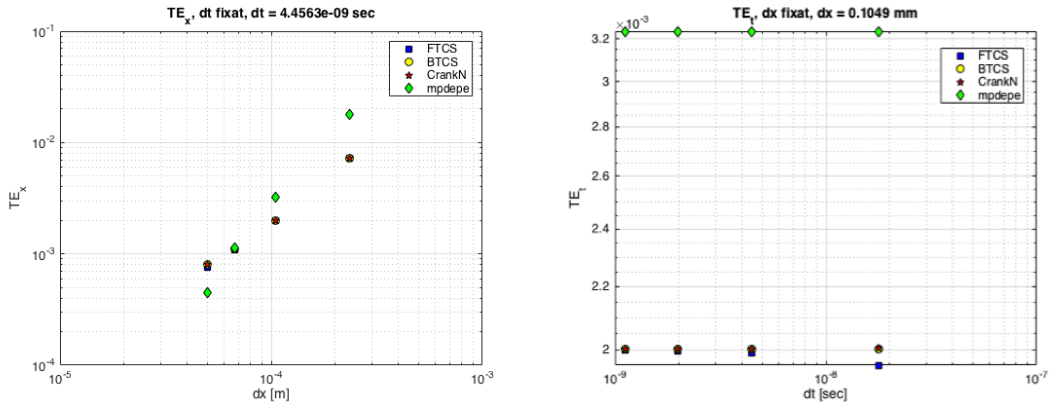


Figure 3.33: The relative errors vs. Δx (left) / vs. Δt (right) (methods: FTCS, BTCS, CN and pdepe); Δt fixed (left), Δx fixed (right).

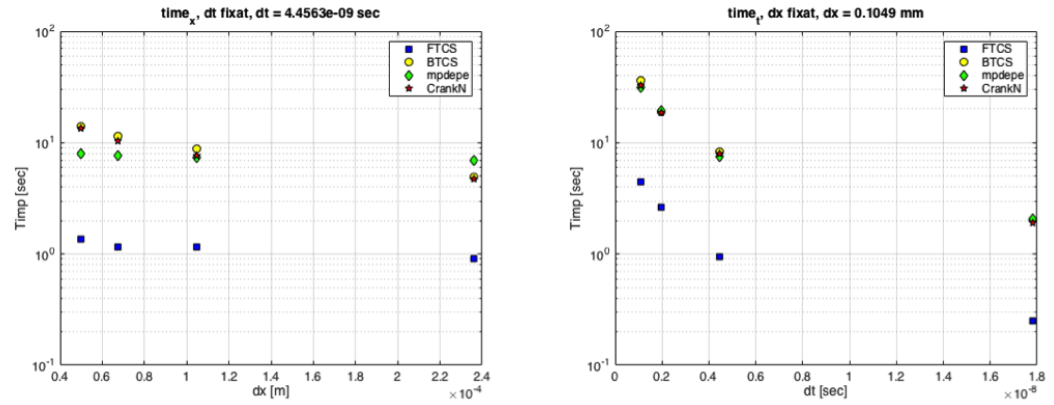


Figure 3.34: Total computing time vs. Δx (left) / vs. Δt (right) (methods: FTCS, BTCS, CN and pdepe); Δt fixed (left), Δx fixed (right).

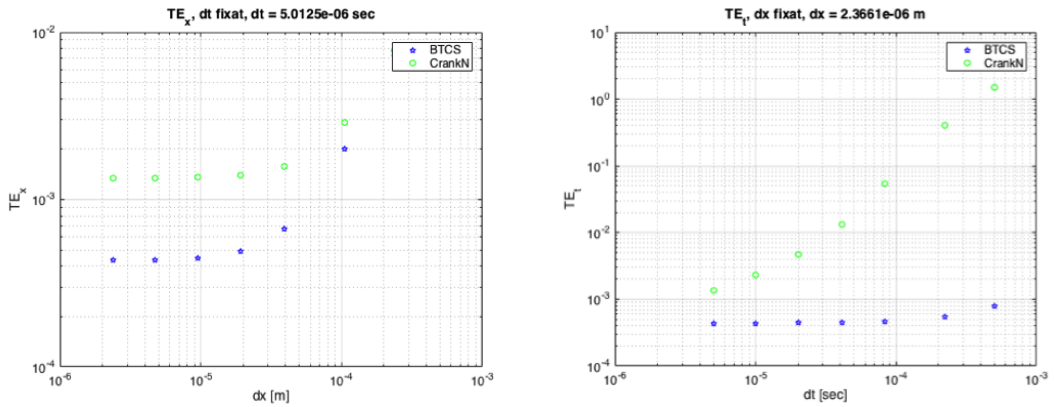


Figure 3.35: The relative errors vs. Δx (left) / vs. Δt (right) (methods: BTCS and CN); Δt fixed (left), Δx fixed (right).

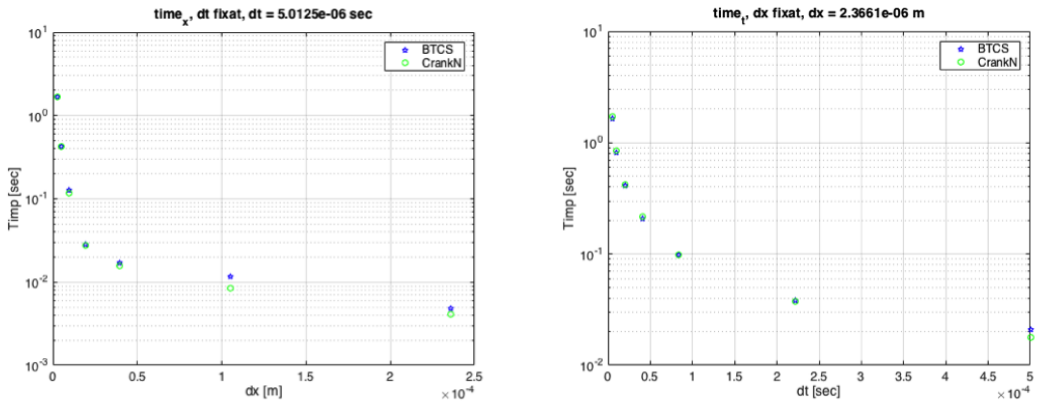


Figure 3.36: Total computing time vs. Δx (left) / vs. Δt (right) (methods: BTCS and CN); Δt fixed (left), Δx fixed (right).

cretization steps are relatively close, the dependence time vs. step does not matter as much as the order of the computing time. The difference in time between the explicit method and the implicit ones is due to the fact that FTCS does not require solving a system of equations at each time point. However, the FTCS method has the disadvantage that it is not stable for time and space steps that do not respect the constraint (3.33).

The BTCS and CN methods are unconditionally stable, which means they can be used for coarser time and space meshes. This allows the comparison between the two implicit methods for time and space steps of comparable size. Figures 3.35 and 3.36 show the relative error and the computing time vs. the discretization step (Δx, Δt) for BTCS and CN, having Δt fixed at $5 \cdot 10^{-6}$ sec (400 nodes) and Δx varying between $[2.366 \cdot 10^{-6}, 1.888 \cdot 10^{-4}]$ m ($5 \div 400$ nodes) and vice versa, having Δx fixed at $2.366 \cdot 10^{-6}$ m (400 nodes) and Δt varying between $[5.013 \cdot 10^{-6}, 4 \cdot 10^{-4}]$ sec ($5 \div 400$ nodes).

It is surprising that for this configuration BTCS has significantly lower errors than CN whereas the computing times are approximately equal.

3.4.7 Efficient implementation in Matlab

The system solved at each time step has the matrix ($\mathbf{A}_n$) tridiagonal, which means that of N^2 total elements, only $3N-2$ are not null. This suggests that the use of Matlab sparse matrix techniques will prove extremely effective. The implementation using sparse matrices assumes that for a matrix only nonzero elements and their positions are stored in memory (triplets: value, row, column). The storing technique influences the shape of the algorithms and therefore the execution speed, as the complexity of the algorithm will depend on the number of non-zero elements and not on the size of the system matrix.

The algorithms presented in the previous section were implemented with the classical technique for full matrices and the sparse matrix technique in Matlab. Figures 3.37 show the computing time dependence for FTCS, BTCS, CN and pdepe methods for the algorithms modified to use sparse matrices (having temporal and spatial discretization steps from Table 3.12). In Figures 3.38 only the computing time of sparse matrices algorithms for the second configuration (coarser mesh) is shown, for which only BTCS

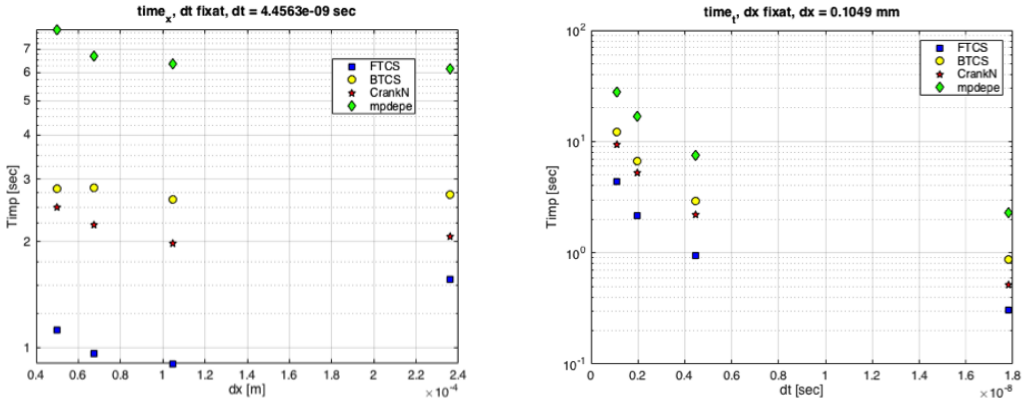


Figure 3.37: Total computing time vs. Δx (left) / vs. Δt (right) (methods: FTCS, BTCS, CN and pdepe); Δt fixed (left), Δx fixed (right); sparse matrices code.

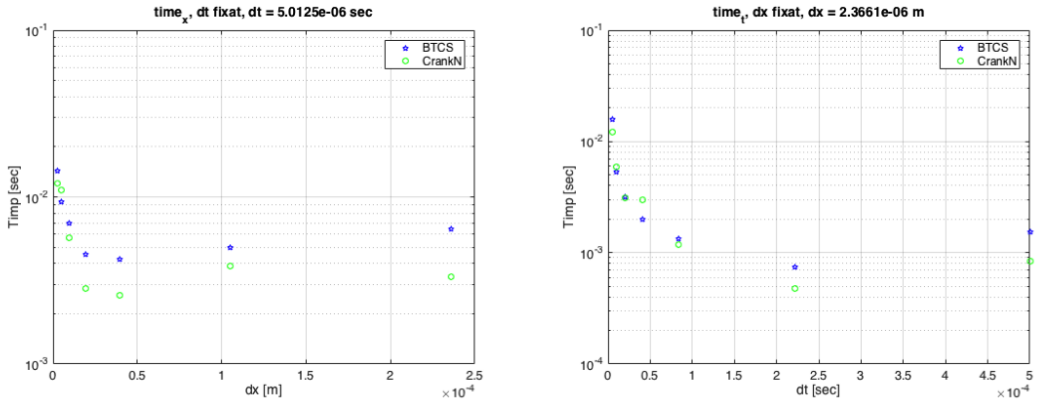


Figure 3.38: Total computing time vs. Δx (left) / vs. Δt (right) (methods: BTCS and CN); Δt fixed (left), Δx fixed (right); sparse matrices code.

and CN are stable.

Following sparse matrix implementation, the numerical results are the same, but the total computing time is significantly lower. Table 3.13 contains the computing times for FTCS, BTCS, CN and pdepe. For the implicit methods it is also given the total time for the second configuration (time and space steps of comparable size), corresponding to Figures 3.38.

As expected, the FTCS method does not show any improvement in computing time, because it does not use a system matrix at every time step. Neither the *pdepe* method has improved because there are no sparse matrices options in its call parameters. In contrast, for BTCS and CN, the improvement of the computing time is significant, about 3 times for the initial configuration (from Table 3.12) and 50 to 120 times for the coarser configurations.

The study of efficient implementation in Matlab, but especially the study of time integration methods is closely related to order reduction and computation time minimization, which becomes relevant when the simulation procedure is incorporated into simulation of large-scale neural networks. Explicit numerical time integration methods such as FTCS are very easy to develop and show small truncation errors but cannot be used for any temporal and spatial configuration due to their instability. Implicit methods such as

Table 3.13: The computing times – full and sparse matrices.

		Full matrices code [s]	Sparse matrices code [s]
FTCS	Δt fixed	4.11	4.15
	Δx fixed	8.05	8.258
BTCS	Δt fixed 2nd config.	**3.16**	**0.06**
	Δt fixed	**36.64**	**12.03**
	Δx fixed 2nd config.	**4.31**	**0.04**
	Δx fixed	**69.86**	**23.1**
CN	Δt fixed 2nd config.	**3.01**	**0.05**
	Δt fixed	**32.90**	**9.88**
	Δx fixed 2nd config.	**4.4**	**0.034**
	Δx fixed	**52.23**	**17.56**
pdepe	Δt fixed	30	30.9
	Δx fixed	61.1	56.5

BTCS or CN are unconditionally stable and have acceptable truncation errors for larger discretization steps. As the system matrix is intrinsically sparse, using sparse matrices implementation in Matlab can reduce the computational effort generated by the solving of systems of equations.

3.5 Conclusion regarding One-dimensional Models for Neuronal Signals' Transmission

This chapter was dedicated to the one-dimensional model for neuronal signals' transmission. The analytical solution and the transmission parameters determined analytically are compared to the data reported in literature. Next, the analytical model is used as a reference, thus validating the numerical model and methods.

The problem is formulated as a field problem (EC+EQS) from which the mathematical equations are deducted, as well as the boundary conditions. As the model is linear the analytical solution is found for the problem previously formulated. The observed behaviour leads to a modification to the original model in order to take into account the effect of ion pumps. Several characteristic parameters are determined for the analytical model: the line attenuation, the maximum transmission length for the signal, the signal delay and the transmission speed. An analytical expression validated by the literature for the transmission speed is deducted (*"the speed of thought"*).

The numerical model is obtained by discretization of the model's equations. The stationary potential along the line is obtained using dual spatial networks, whose values frame the analytical values of the potential. As it is based on discretization, the numerical model's accuracy depends on the meshing step. This dependence is determined for different discretization methods of the boundary conditions. The nodal matrix $\mathbf{A}_{\mathrm{n}}$ of the equivalent electric circuit (tridiagonal and symmetric) is determined from the coefficients matrix $\mathbf{A}_1$.

In order to verify the assumption that the transmission parameters previously determined depend on the form of the excitation signal, four neural signals were chosen from the literature in order to replace the initial excitation signal, obtained with the difference of two exponentials. The standard neural signals selected appeared in the form of curves in figures, so they had to be digitized in order to be used in the numerical procedure.

Numerical time integration such as FTCS, BTCS and Crank-Nicholson are used to find the variation of the solution over time. They are compared in terms of accuracy (relative errors), stability and efficiency (computation time). The last part of this chapter stresses the importance of efficient implementation of the algorithms. The use of sparse matrices implementation in Matlab reduces the computational effort generated by the solving of systems of equations at every time step in numerical integration.

Chapter 4

Reduced Order Models of Myelinated Axonal Compartments

This chapter is dedicated to the order reduction of models for myelinated axonal compartments. Model reduction consists in the approximation of a high complexity input/output system with a lower complexity one, which describes the relation between the input and the output signals with acceptable differences. In the time domain the complexity is given by the number of state variables and in the frequency domain by the number of poles of the transfer function. Models of very high complexity are generated for the analysis of complicated installations and devices, or after discretizing with numerical methods the PDEs of physical fields. Considering that such models have thousands, tens of thousands, hundreds of thousands, and even millions of degrees of freedom, the objective is that the model extracted from the reduction to be simulated without losing the essence of the model with minimal computation effort, by reducing the complexity order to a few hundred or even dozens of units.

Order reduction methods have been developed in the systems theory, applied mathematics, numerical methods and circuit theory, following the requirements of designers of large integrated electronic circuits (VLSIs), which needed to model the capacitive and inductive parasitic effects and high speed interconnection lines. The aim was to simulate the models of these devices as efficiently as possible within reasonable accuracy bounds for design validation and potential optimal redesign. Meanwhile, reduction methods applied to structural analysis have been developed, in the study of mechanical structures or devices, of complex installations, in aerospace and even in biology. Books, collections of articles, doctoral theses and thousands of articles have been dedicated to this field, which proves the importance and timeliness of this topic. This chapter deals with model reduction in computational neuroscience, particularly in modeling the signal transmission in neuronal circuits.

Three classes of models are considered in this chapter, either with distributed parameters (2.5D EQS–ElectroQuasiStatic, 1D TL-Transmission Lines) or with lumped parameters (0D). The bio-multi-physics procedure is applied once more, every model being systematically analyzed from the conceptual, mathematical, analytical and numerical points of view, and finally being reduced to low-order models with different reduction procedures

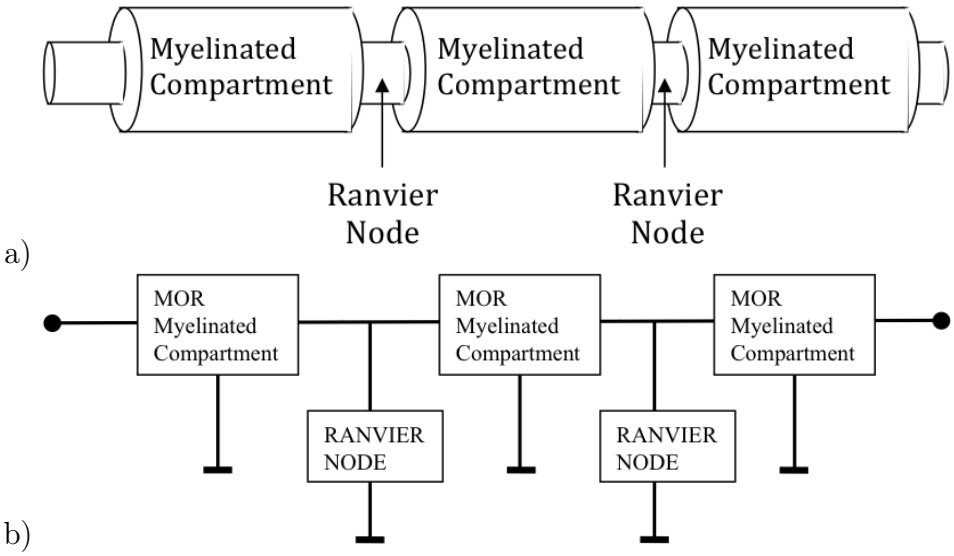


Figure 4.1: The myelinated axon; a) simplified geometrical model; b) circuit mapping.

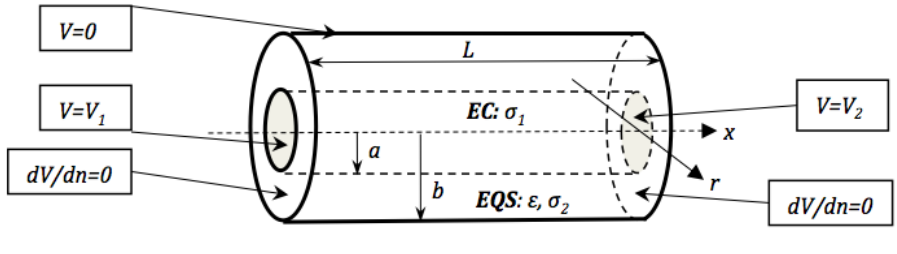


Figure 4.2: Simplified geometrical model of the myelinated compartment.

and error control (which is closely connected to the reduced model order and complexity). The main goal of this part of research is to identify the best procedure for order reduction of each case. An appropriate error estimator is proposed in order to assess the accuracy of the models. This is the foundation of a procedure able to find the simplest reduced model having an imposed precision.

This section is a development of the ideas presented in [66].

4.1 Axisymmetric (2.5D) models

4.1.1 Conceptual 2.5D model

The conceptual modeling of the axon starts from the simplified geometrical model presented in Fig. 4.1, from which the model of a cylindrical myelinated compartment is extracted (Fig. 4.2). The model is described by the following geometrical parameters: the cytoplasm radius – a, the outer radius of the compartment – b, the length of the myelinated compartment – L.

A reference test case, with the following values is considered:

$a = 7\,\mu$m; $b = 10\,\mu$m; $L = 0.25\lambda_0$ or λ_0 or $4\lambda_0$, $\lambda_0 = 223\,\mu$m;

$\sigma_1 = 1.0824$ S/m; $\sigma_2 = 2.04 \cdot 10^{-4}$ S/m; $\varepsilon = 15.44 \cdot \varepsilon_0$.

The model is analyzed in the frequency range $f_m \leq f \leq f_M$, with the limits $f_m = 10^3$ Hz, and $f_M = 10^7$ Hz.

By carrying out a dimensional analysis of the material constants σ, ε, μ, the following characteristic times of the electromagnetic phenomena can be defined [160]:

- $\tau_e = \varepsilon/\sigma$, the relaxation time of the charge and thus of the electric field in a conductor;
- $\tau_m = \mu\sigma L^2$, the diffusion time of the current and thus of the magnetic field in a conductor;
- $\tau_{em} = L/c$, having $c^2 = 1/(\varepsilon\mu), \tau_{em}^2 = \tau_e\tau_m$, the time in which an electromagnetic wave with velocity c travels along the line of length L.

If we use the characteristic time τ (defined as duration, period, or time constant) to describe the speed of the phenomena analyzed, then it is possible to distinguish between rapid and slow EM field regimes. By considering the minimal and maximal values of the five characteristic parameters (σ, ε, μ, L, τ), we are able to generate the map of EM field regimes function of characteristic times (Fig. 4.3). As a point is closer to the left boundary of the domain, the modeling error is higher, being inversely proportional to the distance to the border. The map in Fig. 4.3 indicates that the cytoplasm is operating in the ElectroConduction (EC) region, whereas the membrane has to be modeled with ElectroQuasiStatic (EQS) field.

Consequently, within the cytoplasm, of conductivity σ_1, an EC field regime is considered, while the myelinated layer, between $r = a$ and $r = b$, of conductivity σ_2 and permittivity ε, is considered to be in the EQS regime. In more accurate models, the myelinated layer is modeled as an anisotropic domain, in which the conductivity σ_2 is a tensor, principal directions being radial (σ_2') and axial (σ_2'').

4.1.2 Mathematical 2.5D model

In both domains of the compartment model the electric field is irrotational and thus an electric scalar potential can be defined. Due to the axial symmetry of the domain, this potential depends only on two spatial coordinates, the radius r and the axial position x. We will denote by V_1 the potential that corresponds to the cytoplasm and by V_2 the potential that corresponds to the membrane:

$$V(r,x) = \begin{cases} V_1, & \text{for } 0 < r < a. \\ V_2, & \text{for } a < r < b. \end{cases}$$

The field equations in the two sub-domains are:

$$\text{EC} : \begin{cases} \text{div}\mathbf{J} = 0 \\ \text{curl}\mathbf{E} = 0 \Rightarrow \mathbf{E} = -\text{grad}V_1 \Rightarrow \Delta V_1 = 0 \\ \mathbf{J} = \sigma_1\mathbf{E}. \end{cases} \tag{4.1}$$

a)

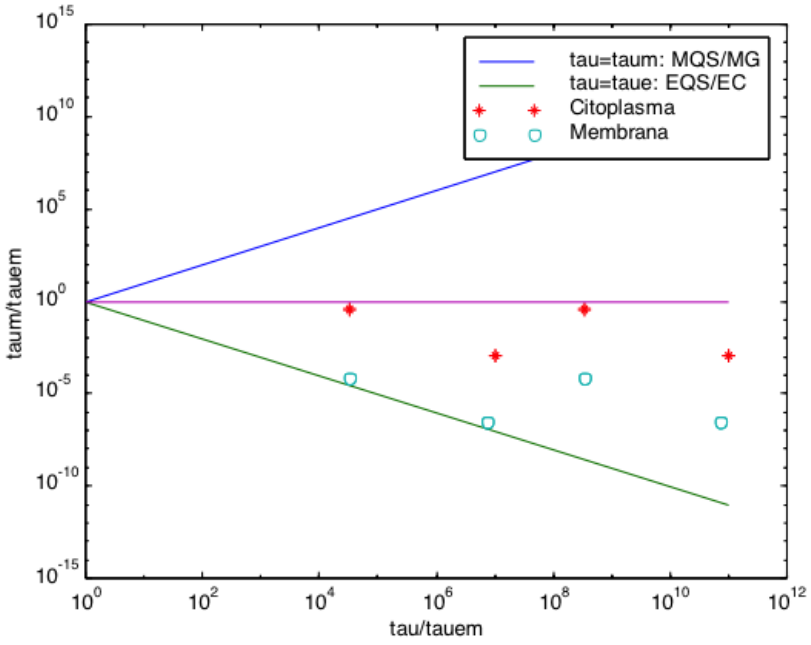


Figure 4.3: Map of electromagnetic field regimes for membrane and cytoplasm of the myelinated axon.

$$\text{EQS}: \begin{cases} \text{div}\mathbf{J} = -\partial\rho/\partial t \\ \text{div}\mathbf{D} = \rho \\ \text{curl}\mathbf{E} = 0 \Rightarrow \mathbf{E} = -\text{grad}V_2 \Rightarrow \Delta V_2 = 0 \\ \mathbf{J} = \sigma_2\mathbf{E}, \mathbf{D} = \varepsilon\mathbf{E}, \end{cases} \tag{4.2}$$

where ρ is a generalized function of space. Therefore, $\Delta \mathrm{V}_2(s) = 0$ (harmonic) in each subdomain, but not on the entire computing domain since on the discontinuity surface $r = a$ the following interface conditions are met:

$$\begin{gathered} V_1(a,x,t) = V_2(a,x,t) \Rightarrow \mathrm{V}_1(a,x,s) = \mathrm{V}_2(a,x,s); \\ \sigma_1 \frac{\partial \mathrm{V}_1}{\partial n} = (\sigma_2 + \varepsilon s)\frac{\partial \mathrm{V}_2}{\partial n}. \end{gathered} \tag{4.3}$$

The solution $V(x,r,t)$ for $0 < x < L, 0 < r < b, 0 < t < t_{max}$ satisfies mixed boundary conditions:

$$\begin{cases} V(0,r,t) = V_1, & 0 < r < a; \\ V(L,r,t) = V_2, & 0 < r < a; \\ V(x,b,t) = 0, & 0 < x < L; \\ \mathrm{d}V(0,r,t)/\mathrm{d}n = 0, & a < r < b; \\ \mathrm{d}V(L,r,t)/\mathrm{d}n = 0, & a < r < b \\ \text{and } V(x,r,0) = 0. \end{cases} \tag{4.4}$$

In reality, due to the ion pumps, in resting state (which in the formulation above would affect the initial condition) the potential has non-null value $V_0 = -70$ mV. We carried out the study for variations of the potential from this state of equilibrium, so the obtained dynamical system to be modeled and reduced is linear and not affine. These

boundary conditions define a linear multipolar Electric Circuit Element (ECE, [161]) with distributed parameters, with three terminals, one being the ground and the other two voltage-controlled. Its dynamic behavior is fully described by the symmetrical matrix of operational admittances $\mathbf{Y}$, which has only two independent elements: the input admittance $\mathrm{Y}_{11}(s) = \mathrm{Y}_{22}(s)$ and the transfer admittance $\mathrm{Y}_{12}(s) = \mathrm{Y}_{21}(s)$, both complex functions of complex frequency s:

$$\begin{aligned} &\mathbf{I}(s) = \mathbf{Y}(s)\mathbf{V}(s), \\ &\text{with } \mathbf{I}(s) = \begin{bmatrix} \mathrm{I}_1 \\ \mathrm{I}_2 \end{bmatrix}; \mathbf{V}(s) = \begin{bmatrix} \mathrm{V}_1 \\ \mathrm{V}_2 \end{bmatrix}; \mathbf{Y}(s) = \begin{bmatrix} \mathrm{Y}_{11} & \mathrm{Y}_{12} \\ \mathrm{Y}_{21} & \mathrm{Y}_{22} \end{bmatrix}. \end{aligned} \tag{4.5}$$

Here the input signals are the Laplace transforms of the terminals' potentials: $\mathrm{V}_1(s) = \mathcal{L}[V(0,0,t)]$, $\mathrm{V}_2(s) = \mathcal{L}[V(L,0,t)]$ and the output signals are the terminals' currents: $\mathrm{I}_1(s) = \mathcal{L}[I_1(t)]$, $\mathrm{I}_2(s) = \mathcal{L}[I_2(t)]$.

If the circuit element is excited in current, then the impedance matrix $\mathbf{Z} = \mathbf{Y}^{-1}$ describes it [66].

4.1.3 Analytical 2.5D model

If the boundary conditions are slightly modified, so that on $x = 0$ and $x = L$ Neumann boundary conditions are set:

$$\begin{cases} \frac{\mathrm{d}V}{\mathrm{d}x}\Big|_{x=0} = \frac{I_1(t)}{\pi a^2 \sigma_1}, & 0 < r < a; \\ \frac{\mathrm{d}V}{\mathrm{d}x}\Big|_{x=L} = \frac{I_2(t)}{\pi a^2 \sigma_1}, & 0 < r < a; \\ V(x,b,t) = 0, & 0 < x < L; \\ \frac{\mathrm{d}V}{\mathrm{d}x}\Big|_{x=0} = 0, & a < r < b; \\ \frac{\mathrm{d}V}{\mathrm{d}x}\Big|_{x=L} = 0, & a < r < b; \\ V(x,r,0) = 0, \end{cases} \tag{4.6}$$

the solution can be analytically determined, using the separation of variables. This change introduces a modeling error, due to the assumption that the current is uniformly distributed on each terminal, and therefore the terminal is no longer strictly equipotential. In reality, it is very likely that neither the potential nor the current density be perfectly constant on the terminal. The simplifying hypothesis of ignoring the radial variation of the current, which allows the derivation of an analytical solution, is acceptable, as this variation is expected to be small. The obtained radial variation of the potential can be used to compute an error estimator for the analytical method applied to this model called "non-ECE" in what follows. Since is excited in current, it will be characterized by an impedance matrix $\mathbf{Z} = \mathbf{Y}^{-1}$. The input impedance Z_{11} of the near terminal is computed as the ratio between the central potential $V(0,0,s)$ and the injected current $\mathrm{I}_1(s)$ under the assumption that the current of the far end terminal is null. The transfer impedance Z_{21} is the ratio of the potential at the center of the far end terminal $V(L,0,s)$ and the current $\mathrm{I}_1(s)$ injected into the near terminal, under the condition that the far end terminal current is null ($I_2 = 0$).

The equation $\mathrm{div}(\sigma \mathrm{grad} V) = 0$, satisfied by the potential V in each homogeneous

subdomain, has the following form in cylindrical coordinates:

$$\operatorname{div}(\sigma \operatorname{grad} V) = \frac{1}{r}\frac{\partial}{\partial r}\left(r\sigma\frac{\partial V}{\partial r}\right) + \frac{\partial}{\partial x}\left(\sigma\frac{\partial V}{\partial x}\right) = 0, \tag{4.7}$$

where the axial symmetry of the function (i.e. independence on the azimuthal angle) was taken into account. In each homogeneous subdomain the potential V is a harmonic function, a solution of the Laplace equation. According to the separation of variables method [9], in each homogeneous subdomain the potential V is assumed to have the form:

$$V(x, r) = X(x)R(r). \tag{4.8}$$

Substituting (4.8) in (4.7) it follows that the PDE can be decomposed in two linear ODEs satisfied by the two functions X and R:

$$\begin{aligned} X\frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}r}\left(\sigma r\frac{\mathrm{d}R}{\mathrm{d}r}\right) + \mathrm{R}\sigma\frac{\mathrm{d}^2\mathrm{X}}{\mathrm{dx}^2} = 0 & \\ \Longrightarrow X\frac{1}{r}\left(\sigma r R'\right)' + \sigma R x'' = 0 \quad | : (XR\sigma) & \\ \Longrightarrow -\frac{1}{r}\frac{\left(\sigma r R'\right)'}{\sigma R} = \frac{X''}{X}. & \end{aligned} \tag{4.9}$$

The left side of this equality is only dependent on r, whereas the right side is dependent only on x. We can therefore deduce that both terms are constant and equal with a positive real constant λ^2, called constant of separation:

$$\begin{cases} \frac{X''}{X} = \lambda^2 \\ -\frac{(\sigma r R')'}{\sigma r R} = \lambda^2 \end{cases}. \tag{4.10}$$

The first is an ODE with constant coefficients having real characteristic roots ($\pm\lambda$). Consequently, its solution is a linear combination of $e^{\lambda x}$ and $e^{-\lambda x}$:

$$X(x) = A\,\mathrm{sh}(\lambda x) + B\,\mathrm{ch}(\lambda x) \tag{4.11}$$

where A and B are integration constants.

In (4.10) σ is piecewise constant (σ_1 for $0 < r < a$ and σ_2 for $a < r < b$). Therefore, on each homogeneous subdomain, the function R is the solution of the differential equation:

$$\begin{aligned} -R' - rR'' = \lambda^2 rR \quad | \cdot r & \\ \Longrightarrow r^2R'' + rR' + \lambda^2 r^2 R = 0, & \end{aligned} \tag{4.12}$$

which is a combination of zero order Bessel functions with the following form for the general solution:

$$R(r) = CJ_0(\lambda x) + DY_0(\lambda x). \tag{4.13}$$

In conclusion, the general solution of the Laplace equation $V(r, x)$ has the form:

$$V(\mathrm{r}, \mathrm{x}) = (CJ_0(\lambda x) + DY_0(\lambda x)(A\,\mathrm{sh}(\lambda x) + B\,\mathrm{ch}(\lambda x)). \tag{4.14}$$

The parameter λ and the integration constants are derived by imposing the boundary conditions:

- on terminal 2 ($x = L$, $0 < r < a$)

$$\left.\frac{\partial V}{\partial n}\right|_{x=L} = 0 \Rightarrow \left.\frac{\partial V}{\partial x}\right|_{x=L} = 0 \Rightarrow \left.\frac{\partial X}{\partial x}\right|_{x=L} = 0$$
$$\Longrightarrow A\lambda \operatorname{ch}(\lambda L) + B\lambda \operatorname{sh}(\lambda L) = 0$$
$$\Longrightarrow B = \frac{-A \operatorname{ch}(\lambda L)}{\operatorname{sh}(\lambda L)}.$$

We derive the new form for $X(x)$ as:

$$X(x) = A\operatorname{sh}(\lambda x) - A\frac{\operatorname{ch}(\lambda L)}{\operatorname{sh}(\lambda L)}\operatorname{ch}(\lambda x) = \frac{A(\operatorname{sh}(\lambda x)\operatorname{sh}(\lambda L) - \operatorname{ch}(\lambda x)\operatorname{ch}(\lambda L))}{\operatorname{sh}(\lambda L)}$$
$$\Longrightarrow X(x) = -\frac{A\operatorname{ch}(\lambda(x-L))}{\operatorname{sh}(\lambda L)}.$$

By making the notation $B' = -\frac{A}{\operatorname{sh}(\lambda L)}$ and using the fact that cosh is an even function we get that:

$$X(x) = B' \operatorname{ch}(\lambda(L-x)). \tag{4.15}$$

$V(0,x)$ needs to be finite, so in the first subdomain ($0 < r < a$) D has to be 0 and thus:

$$V(r,x) = CJ_0(\lambda x)\operatorname{ch}(\lambda(L-x)), \tag{4.16}$$

where $CB' = C$.

In the second subdomain ($a < r < b$) the solution also contains the Bessel function Y, because it does not include the axis:

$$R(r) = \begin{cases} C_1 J_0(\lambda r), & 0 < r < a \\ C_2 J_0(\lambda r) + D_2 Y_0(\lambda r), & a < r < b \end{cases}. \tag{4.17}$$

After renaming the constants above, the potential has the general form:

$$V(r,x) = \begin{cases} BJ_0(\lambda r)\operatorname{ch}(\lambda(L-x)), & 0 < r < a \\ (CJ_0(\lambda r) + DY_0(\lambda r))\operatorname{ch}(\lambda(L-x)), & a < r < b \end{cases} \tag{4.18}$$

with $B = B'C$, $C = C_2B'$, $D = D_2B'$.

- on the interface $r = a$:

$$V_1(a,x) = V_2(a,x), \quad 0 < x < L;$$
$$\frac{\partial \underline{V}_1}{\partial n}(\sigma_1 + j\omega\varepsilon_1) = \frac{\partial \underline{V}_2}{\partial n}(\sigma_2 + j\omega\varepsilon_2) \tag{4.19}$$
$$\Longrightarrow \frac{\partial \underline{V}_1}{\partial n} = \frac{\sigma_2 + j\omega\varepsilon_2}{\sigma_1 + j\omega\varepsilon_1}\frac{\partial \underline{V}_2}{\partial n}.$$

With the notation:

$$\beta = \frac{\sigma_2 + j\omega\varepsilon_2}{\sigma_1 + j\omega\varepsilon_1} \tag{4.20}$$

we can write that:

$$\left.\frac{\partial \underline{V}_1}{\partial n}\right|_{r=a} = \beta \left.\frac{\partial \underline{V}_2}{\partial n}\right|_{r=a}. \tag{4.21}$$

In EC regime, $\omega = 0 \Rightarrow \beta = \frac{\sigma_2}{\sigma_1} \in \mathbb{R}$.

In EQS regime, $\omega \neq 0 \Rightarrow \beta \in \mathbb{C}$.

The two expressions of potential (4.18) satisfy on the boundary (4.19) and (4.21):

$$\begin{aligned} &BJ_0(\lambda a) = (CJ_0(\lambda a) + DY_0(\lambda a)) \Rightarrow B = C + D\frac{Y_0(\lambda a)}{J_0(\lambda a)}; \\ &BJ_0'(\lambda a) = (CJ_0'(\lambda a) + DY_0'(\lambda a))\,\beta \end{aligned} \tag{4.22}$$

$$\Rightarrow B = \beta\left(C + D\frac{Y_0'(\lambda a)}{J_0'(\lambda a)}\right) \Rightarrow B = \beta\left(C + D\frac{Y_1(\lambda a)}{J_1(\lambda a)}\right). \tag{4.23}$$

The system consisting of (4.22) and (4.23) has two equations and three unknowns, therefore the system must be rewritten using ratios $\frac{B}{C}$ and $\frac{D}{C}$:

$$\begin{cases} \frac{B}{C} = 1 + \frac{D}{C}\frac{Y_0(\lambda a)}{J_0(\lambda a)} \\ \frac{B}{C} = \beta\left(1 + \frac{D}{C}\frac{Y_1(\lambda a)}{J_1(\lambda a)}\right) \end{cases}. \tag{4.24}$$

By eliminating the constant B from (4.24) it follows that:

$$\frac{D}{C}\frac{\beta}{1-\beta} = \frac{1}{\frac{Y_1(\lambda a)}{J_1(\lambda a)} - \frac{1}{\beta}\frac{Y_0(\lambda a)}{J_0(\lambda a)}}. \tag{4.25}$$

- on the boundary $r = b$:

$$V(b,x) = 0, \forall x \in [0,L] \Rightarrow CJ_0(\lambda b) + DY_0(\lambda b) = 0. \tag{4.26}$$

Eliminating $\frac{D}{C}$ from (4.25) and (4.26) leads to the eigenvalues equation:

$$(1-\beta)Y_0(\lambda b)J_0(\lambda a)J_1(\lambda a) + J_0(\lambda b)\left(\beta Y_1(\lambda a)J_0(\lambda a) - Y_0(\lambda a)J_1(\lambda a)\right) = 0, \tag{4.27}$$

which has an infinite number of solutions $\lambda_k, k = 1, 2, \ldots \infty$.

After finding the solutions of this equation (the proper values λ_k), $\frac{D_k}{C_k}$ is determined from (4.25) and $\frac{B_k}{C_k}$ from (4.24) as:

$$\frac{B}{C} = 1 + \frac{1-\beta}{\beta\frac{Y_1(\lambda a)}{J_1(\lambda a)} - \frac{Y_0(\lambda a)}{J_0(\lambda a)}}\frac{Y_0(\lambda a)}{J_0(\lambda a)} = 1 + \frac{(1-\beta)J_1(\lambda a)Y_0(\lambda a)}{\beta J_0(\lambda a)Y_1(\lambda a) - J_1(\lambda a)Y_0(\lambda a)}. \tag{4.28}$$

The general solution of the problem $V(r,x)$ is obtained by superposition of all possible general forms:

$$V(r,x) = \begin{cases} \sum_k C_k \frac{B_k}{C_k} J_0(\lambda_k r)\,\mathrm{ch}\,(\lambda_k(L-x)), & 0 < r < a \\ \sum_k C_k \left(J_0(\lambda_k r) + \frac{D_k}{C_k} Y_0(\lambda_k r)\right)\mathrm{ch}\,(\lambda_k(L-x)), & a < r < b \end{cases} \tag{4.29}$$

Using the notations $R(r)$ from (4.17) and $X(x)$ from (4.15), it follows that:

$$V(r,x) = \sum_k C_k R(\lambda_k r)\,\mathrm{ch}\,(\lambda_k(L-x)), \tag{4.30}$$

where R are eigenfunctions given by:

$$R(\lambda_k r) = \begin{cases} \frac{B_k}{C_k} J_0(\lambda_k r), & 0 < r < a \\ J_0(\lambda_k r) + \frac{D_k}{C_k} Y_0(\lambda_k r), & a < r < b \end{cases}. \tag{4.31}$$

The constant C_k is computed by imposing the Neumann boundary condition

- at $x = 0$:

$$\left.\frac{\partial V}{\partial x}\right|_{x=0} = -\frac{I_1}{\sigma_1 \pi a^2}\mathrm{h}(a-r), \tag{4.32}$$

where h is the Heaviside function (unit step). The function $f(r)$ can be expanded

into Fourier-Bessel series of eigenfunctions:

$$f(r) = \sum_k \lambda_k C_k R\left(\lambda_k r\right) \mathrm{sh}\left(\lambda_k L\right) = -\frac{I_1}{\sigma_1 \pi a^2} h(a-r) \tag{4.33}$$

$$\Longrightarrow -\frac{I_1}{\sigma_1 \pi a^2} \mathrm{h}(a-r) = \sum_k F_k R\left(\lambda_k r\right) \tag{4.34}$$

with $F_k = \lambda_k C_k \,\mathrm{sh}\left(\lambda_k L\right)$.

Then the expression of C_k is derived as:

$$C_k = \frac{F_k}{\lambda_k \,\mathrm{sh}\left(\lambda_k L\right)}. \tag{4.35}$$

Substituting (4.35) in (4.16), the potential becomes:

$$V(r,x) = \sum_k \frac{F_k}{\lambda_k \,\mathrm{sh}\left(\lambda_k L\right)} R\left(\lambda_k r\right) \mathrm{ch}\left(\lambda_k (L-x)\right). \tag{4.36}$$

The potential on every terminal has the expressions:

– on terminal 1: $x = 0$, $0 < r < a$:

$$V(r,0) = \sum_k \frac{F_k}{\lambda_k \,\mathrm{th}\left(\lambda_k L\right)} R\left(\lambda_k r\right); \tag{4.37}$$

– on terminal 2: $x = L$, $a < r < b$:

$$V(r,L) = \sum_k \frac{F_k}{\lambda_k \,\mathrm{sh}\left(\lambda_k L\right)} R\left(\lambda_k r\right). \tag{4.38}$$

The Fourier coefficients F_k of this series result from the orthogonality property of the eigenfunctions:

$$\left\langle R_0\left(\lambda_j r\right), R_0\left(\lambda_k r\right)\right\rangle = \int_0^b r\sigma(r) R\left(\lambda_j r\right) R\left(\lambda_k r\right) \mathrm{d}r.$$

In EC regime and with $j = k$ this relation becomes:

$$\left\|R_k\right\|^2 = \sigma_1 \int_0^a r \frac{B_k^2}{C_k^2} J_0^2\left(\lambda_k r\right) \mathrm{d}r + \sigma_2 \int_a^b r \left(J_0\left(\lambda_k r\right) + \frac{D_k}{C_k} Y_o\left(\lambda_k r\right)\right)^2 \mathrm{d}r. \tag{4.39}$$

Considering the following:

$$\int_0^a r J_0^2\left(\lambda_k r\right) \mathrm{d}r = \frac{a^2}{2}\left(J_0^2\left(\lambda_k a\right) + J_1^2\left(\lambda_k a\right)\right) \tag{4.40}$$

$$\int_a^b r J_0^2\left(\lambda_k r\right) \mathrm{d}r = \frac{1}{2}\left(b^2\left(J_0^2\left(\lambda_k b\right) + J_1^2\left(\lambda_k b\right)\right) - a^2\left(J_0^2\left(\lambda_k a\right) + J_1^2\left(\lambda_k a\right)\right)\right), \tag{4.41}$$

it follows that:

$$\int_a^b r J_0\left(\lambda_k r\right) Y_0\left(\lambda_k r\right) \mathrm{d}r = \int_{\lambda_k a}^{\lambda_k b} \frac{x}{\lambda_k} J_0(x) Y_0(x) \frac{\mathrm{d}x}{\lambda_k}$$

$$= \frac{1}{2}\left(b^2\left(J_0\left(\lambda_k b\right) Y_0\left(\lambda_k b\right) + J_1\left(\lambda_k b\right) Y_1\left(\lambda_k b\right)\right) - a^2\left(J_0\left(\lambda_k a\right) Y_0\left(\lambda_k a\right) + J_1\left(\lambda_k a\right) Y_1\left(\lambda_k a\right)\right)\right) \tag{4.42}$$

The expression (4.39) becomes [66]:

$$\begin{aligned}\|R_k\|^2 &= \sigma_1 \frac{B_k^2}{C_k^2}\frac{a^2}{2}(J_0^2(\lambda_k a)+J_1^2(\lambda_k a))\\ &+\sigma_2\frac{1}{2}(b^2(J_0^2(\lambda_k b)+J_1^2(\lambda_k b))-a^2(J_0^2(\lambda_k a)+J_1^2(\lambda_k a)))\\ &+\sigma_2\frac{D_k^2}{C_k^2}\frac{1}{2}(b^2(Y_0^2(\lambda_k b)+Y_1^2(\lambda_k b))-a^2(Y_0^2(\lambda_k a)+Y_1^2(\lambda_k a)))\\ &+\sigma_2\frac{D_k}{C_k}(b^2(J_0(\lambda_k b)Y_0(\lambda_k b)+J_1(\lambda_k b)Y_0(\lambda_k b))-\\ &+a^2(J_0(\lambda_k a)Y_0(\lambda_k a)+J_1(\lambda_k a)Y_0(\lambda_k a))).\end{aligned} \tag{4.43}$$

The scalar product between $R(\lambda_k r)$ and $f(r)$ leads to the expression of F_k as follows: $\langle R(\lambda_k r), f(r)\rangle = F_k \|R_k\|^2$

$$\Longrightarrow F_k = \frac{\int_0^b r\sigma(r)R(\lambda_k r)\frac{I_1}{\sigma_1\pi a^2}h(a-r)\mathrm{d}r}{\|R_k\|^2} = \frac{\int_0^a r\sigma_1 R(\lambda_k r)\frac{I_1}{\sigma_1\pi a^2}\mathrm{d}r}{\|R_k\|^2}\mathrm{d}r$$

$$\Longrightarrow F_k = \frac{I_1}{\pi a^2\|R_k\|^2}\int_0^a rR(\lambda_k r)\,\mathrm{d}r = \frac{I_1}{\pi a^2\|R_k\|^2}\int_0^a r\frac{B_k}{C_k}J_0(\lambda_k r)\,\mathrm{d}r$$

$$\Longrightarrow F_k = \frac{I_1}{\pi a\|R_k\|^2}\frac{B_k}{C_k}\frac{1}{\lambda_k}J_1(\lambda_k a). \tag{4.44}$$

In conclusion, the potential has the following values in the electrodes centers:

$$V_1 = V(0,0) = \sum_{k=1}^{m}\frac{F_k}{\lambda_k \operatorname{th}(\lambda_k L)}R(0) = \frac{I_1}{\pi a}\sum_{k=1}^{m}\frac{\left(\frac{B_k}{C_k}\right)^2 J_1(\lambda_k a)}{\|\lambda_k R_k\|^2 \operatorname{th}(\lambda_k L)} \tag{4.45}$$

$$\mathrm{V_2} = \mathrm{V(0,L)} = \sum_{k=1}^{m}\frac{F_k}{\lambda_k \operatorname{sh}(\lambda_k L)}R(0) = \frac{I_1}{\pi a}\sum_{k=1}^{m}\frac{\left(\frac{B_k}{C_k}\right)^2 J_1(\lambda_k a)}{\|\lambda_k R_k\|^2 \operatorname{sh}(\lambda_k L)} \tag{4.46}$$

They define the element's impedances $\mathrm{Z_{11}} = \mathrm{V_1/I_1}$; $\mathrm{Z_{12}} = \mathrm{V_2/I_1}$. The Fourier-Bessel series are truncated to m terms in MATLAB. The radial variation of the potential on the near and far electrodes is shown in Fig. 4.4. This variation is relatively negligible for the near electrode $\varepsilon_\infty = \varepsilon_m = (V(0,0)-V(a,0))/V_1 = (1.66537-1.66493)/1.66537 = 2.6\cdot 10^{-4} = 0.0026\%$.

The computational estimation of the analytical value is not affected only by this error but also by the method error of truncating the series and its approximation with a finite sum. Numerical tests show that the series (4.46) corresponding to the far end electrode has an exponential convergence, much faster than that corresponding to the near electrode (4.45). Table 4.1 shows the partial sums and the last term added to the sum of the analytical computation of potential V(0,0). Note that even from the first term six significant digits are correct, and the fourth term adds another significant digit. The truncation error (computed as $\varepsilon_t = |V_m|/V_1$, where V_m is the $\mathrm{m^{th}}$ term added to the sum) is so small $\varepsilon_t = V_4/V_1 = 6.13\cdot 10^{-6} \ll \varepsilon_m$, that the error of the analytical method $\varepsilon_a = \varepsilon_t + \varepsilon_m$ is given by the method error. Therefore, the summation of more than 2 terms in the series is useless, since the gain in accuracy is covered by the method error [66]. The MATLAB code that solves this problem analytically is found in Appendix A2.

Table 4.1: The convergence of the series for impedance Z_{11}.

m	Partial sum	Last term added
1	**1.66536**61913809199e+06	1.6653661913809199e+06
4	**1.665365**7945109792e+06	7.8181849318044057e+00
10	**1.665365**9660544212e+06	6.4287806450348339e-01

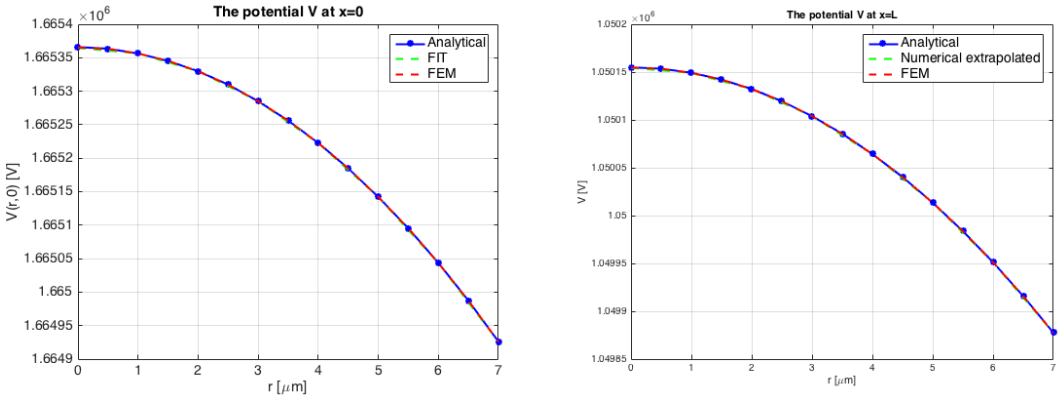


Figure 4.4: The radial variation of the potential at $x = 0$ (left) and $x = L$ (right), computed analytically and numerically (FIT and FEM).

4.1.4 Numerical 2.5D model

The EC field problem was solved with an in-house Finite Integration Technique (FIT) code developed in MATLAB and with FEM (COMSOL) with three variants of boundary conditions: ECEv (ECE with voltage controlled equipotential terminals), ECEc (current controlled equipotential terminals), nonECEc (current controlled terminals, not equipotential, on which Neumann BCs are imposed).

In order to have a relevant comparison we have used for FIT and FEM the same discretization grid/mesh, regular and orthogonal, with quadrilateral FEM cells. Both FIT and FEM give in the case of the nonECEc boundary conditions very similar results with the analytical solution, as shown in Fig. 4.4, where the three curves overlap, which validates both numerical methods. The relative difference of $V(0, 0)$ of FEM from the analytical solution is $5 \cdot 10^{-7}$. The values extracted from the ECE field problem have six significant digits common to those extracted from the nonECE field problem, which shows that the nonECE method error compared to ECE is even lower than $\varepsilon_m = 0.0026\%$ estimated earlier.

The advantage of numerical methods is that they also allow the problem solving with ECE BCs. The difference between FIT and FEM solutions is as small as in the case of the nonECEc BCs. The two Dirichlet boundary conditions ECEv and ECEc give almost identical results, with 10 identical decimals for FEM, and 12 for FIT, respectively, which shows as expected that the excitation type (voltage or current) is not relevant in the ECE models. The final results of the FIT and FEM numerical models were obtained by using component-wise Richardson extrapolation, of the results obtained for two discretization steps (0.5 μm and 1 μm). The errors obtained lead us to consider the Richardson extrapolation for FEM as reference for computing the method errors (Table 4.2).

Table 4.2: Extrapolated values of conductances matrices (Y_{11} and Y_{12}) and resistances matrices (Z_{11} and Y_{12}) in d.c., for FIT and FEM, with 3 BCs.

FEM-extrapolated	ECEv	ECEc	nonECEc
$G_{11} = Y_{11}(0)$	3.055380119410954e-6	3.055380119435735e-6	3.055379595174662e-6
$G_{12} = Y_{12}(0)$	-2.955775083424355e-6	-2.955775083450317e-6	-2.955775163682430e-6
$R_{11} = Z_{11}(0)$	5.103005195720960e+6	5.103005195779775e+6	5.103035666943246e+6
$R_{12} = Z_{12}(0)$	4.936647820764425e+6	4.936647820824645e+6	4.936678279699712e+6
FIT-extrapolated			
$G_{11} = Y_{11}(0)$	3.055380366776102e-6	3.055380366775098e-6	3.055620224050163e-6
$G_{12} = Y_{11}(0)$	-2.955775284972939e-6	-2.955775284973135e-6	-2.956014815660698e-6
$R_{11} = Z_{11}(0)$	5.103002880458375e+6	4.936645517927767e+6	5.103002880519714e+6
$R_{12} = Z_{11}(0)$	4.936645517989167e+6	5.102979781604875e+6	4.936635698549963e+6
Rel. diff. between extrap. values		$\|\mathbf{G}_{\mathrm{FIT}} - \mathbf{G}_{\mathrm{FEM}}\|/\|\mathbf{G}_{\mathrm{FEM}}\|$	
$\mathbf{G} = \mathbf{Y}(0)$	7.4680e-8	7.4686e-8	7.9980e-5
$\mathbf{R} = \mathbf{Z}(0)$	4.5998e-7	4.5999e-7	9.8077e-6

Table 4.3: The values of the constants c and p in the estimation $\varepsilon_{\mathrm{rel}} = c \cdot h^p$.

$err_{rel} = c \cdot h^p$	ECE(v or c)	nonECE (c)
FIT	c = 4535e+3	c = 2.96e+14
	p = 1.95	**p = 2.96**
FEM	**c = 39e+3**	c = 14067e+3
	p = 2.00	p = 2.11

The relative error is approximated by an expression of type $c \cdot h^p$, where c and p are constants that describe the convergence rate and h is the discretization step. Considering two levels of mesh refinement, each halving the step h comparing to the precedent, the convergence rate p was derived. Table 4.3 holds the approximated constants c and p for FIT and FEM, with two different boundary conditions, ECE and nonECE. We can see that both FIT and FEM have a quadratic convergence order, FEM in the case of ECE having a relative error lower than FIT with about 2 orders of magnitude. In theory, quadratic FEM should have a higher convergence rate, but in practice it is limited because the boundary condition is not smooth at all being a non-continuous function [162]. Consequently, the reference values for the extracted circuit functions were given by the Richardson extrapolation of the FEM solution, for $p = 2$. The resulted values seem to have at least 8 exact significant figures. A better reference value may be obtained by using FEM with an adaptive mesh refinement.

Details on numerical solution with FEM.

The two subdomains with axial symmetry have rectangular shape in the coordinates (r, x) and start from radius *eps*: D1 = $(eps, a) \times (0, L)$; D2 = $(a,\ b) \times (0, L)$. They were meshed with a regular grid of squares with a step of $0.5\,\mu$m. For the axon radius of

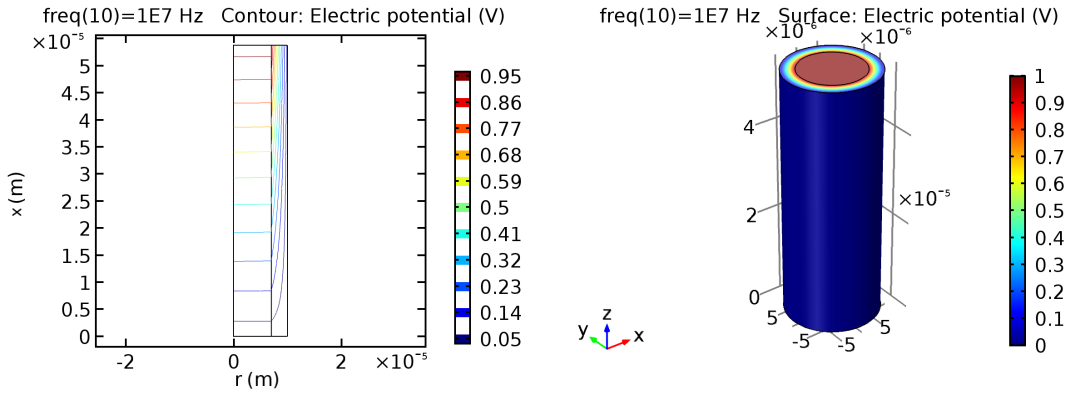


Figure 4.5: Left: Contour plots of the electric potential (FEM). The scales on the at the surface of the computing axis are not equal. Right: The variation of the potential at the surface of the computing domain (FEM). The 3D image is obtained by rotating the 2D solution.

$b = 10\,\mu$m, the radial mesh has (nxa+nxb-1) = 15+7-1 = 21 nodes and for longitudinal length $L = 0.25 \cdot \lambda_0 = 215\,\mu\text{m}/4 = 54\,\mu$m, the mesh has nx = 109 nodes, so a total of n = 2289 nodes. The mesh contains 2230 quadrilateral elements and $2 \times 2289 - 109 - 21 = 4448$ edges.

In FEM, the base functions (also known as trial functions) for potential are second-order Lagrange polynomials, making each cell to have 9 degrees of freedom associated with the four vertices, edges' centers and cell center. The weak form of the equation solved is as follows: find $v \in H_D = \{v \in L^2(D) | \nabla v \in [L^2(D)]^3, v(S_D) = f_D\}$, where D = D1 ∪ D2. The affine Sobolev space, satisfying Dirichlet BC, so that:

$$\begin{aligned} a(v,u) &= 2\pi \int_D \mathbf{J} \cdot \mathbf{E} r \mathrm{dA} \\ &= \int_0^b \int_0^L (\sigma + \varepsilon s)(\nabla v \cdot \nabla u) r \mathrm{d}x \mathrm{d}r \end{aligned} \tag{4.47}$$

is null for any u having Dirichlet null BC. The number of DOFs is 9837, about four times larger than the number of nodes, as we expected. The use of second order elements makes the solution more precise than in the case of the first order, where the number of unknowns is the number of floating nodes (i.e. the inner ones plus those on the Neumann boundary, so that those with essential boundary conditions (Dirichlet) with known potential are excluded.

The boundary conditions are as follows: null Dirichlet for $r = b$, $r < a$ and $x = L$, non-null Dirichlet of value $V_0 = 1$ for $r < a$ and $x = 0$ (essential), null Neumann in rest (natural). The system of linear algebraic equations obtained from $a(u, v) = 0$ by replacing u and v with their expansion in trial functions was solved with a direct method in 5 sec on a two-core MacOS system. The solution is graphically represented in Fig. 4.5.

The EQS analysis in the frequency range [1 kHz, 10 MHz] gives the frequency characteristics shown in Fig. 4.6.

Details on numerical solution with FIT.

The computing domain has been discretized with a mesh similar to that used in FEM. In stationary d.c. mode, each cell $(r_1, r_2) \times (z_1, z_2)$ contributes to the equations system

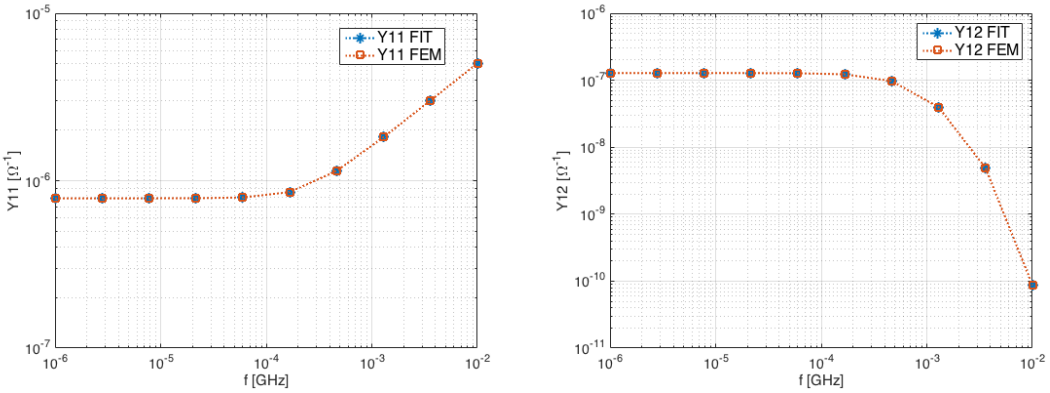


Figure 4.6: The variation of Y_{11} and Y_{12} with frequency (FIT and FEM).

with four conductances, placed on its four edges, having the expressions:

$$\begin{cases} G_{r1} = \frac{\pi\sigma((r_1+\Delta r_1)^2 - r_1^2)}{2\Delta z} \\ G_{r2} = \frac{\pi\sigma(r_2^2 - (r_2-\Delta r_2)^2)}{2\Delta z} & \text{for axial oriented} \\ G_{z1} = G_{z2} = \frac{\pi\sigma\Delta z}{\ln(r_2/r_1)} & \text{for radial oriented} \end{cases} \tag{4.48}$$

where the cell dimensions are: $\Delta z = z_2 - z_1$; $\Delta r = r_2 - r_1$; $\Delta r_1 = \Delta r_2 = \Delta r/2$; $G_{z1} = G_{z2} = 0$, if $r_1 = 0$.

The simplest implementation method in MATLAB uses the nodal technique to write the equations of this resistive electric circuit: $(\mathbf{AGA}^{\mathrm{T}})\mathbf{V} = \mathbf{J}$, where $\mathbf{A}$ is the edges-nodes incidence matrix, $\mathbf{G}$ is the diagonal matrix of conductances (4.48) and $\mathbf{J}$ is the array of injected currents into nodes.

The code is executed extremely fast since it does not contain *for* loops, but makes use of sparse matrices. The implementation is simplified if the nodes are numbered systematically from 1 to N, for example starting along the axis and then similarly for increasing radius, so that going through the nodes $i = 1 : nr - 1$ and $j = 1 : nz - 1$, all cells $k = (i-1)nz + j$ are covered. If the edges are numbered in agreement with the nodes, for example the axial ones having double indices $(2k)$ than the initial node (k) and the radial ones an uneven index $(2k+1)$, a total of $2N$ edges is obtained. Thus virtual edges are introduced on one side of the boundary, which are removed eventually. They correspond to null columns in matrix $\mathbf{A}$. By splitting the node set into those floating, with unknown potentials $(\mathbf{V_1})$ and those placed on the Dirichlet boundary, with known potentials $(\mathbf{V_2})$ and considering the corresponding split of nodal conductances matrix:

$$\begin{cases} \mathbf{G_{11}V_1} + \mathbf{G_{11}V_2} = \mathbf{J_1} \\ \mathbf{G_{21}V_1} + \mathbf{G_{22}V_2} = 0 \end{cases} \Rightarrow \mathbf{G_{11}V_1} = \mathbf{J_1} - \mathbf{G_{11}V_2} \tag{4.49}$$

leads to the system of linear equations that is solved to determine the unknown potentials $(\mathbf{V_1})$. Actually, this technique is also used in FEM to handle essential boundary conditions.

Under nonECEc boundary conditions, the injected current on the terminal is known, so the vector $\mathbf{J}$ is no longer null. The current injected into a circular crown between two successive nodes can be assigned to the initial node (case v1 in Fig. 4.7), to the final node (case v2), or the current injected into the node is calculated using the dual network edges as in dFIT [163] (option 3). The results are presented comparatively in Fig. 4.7. The

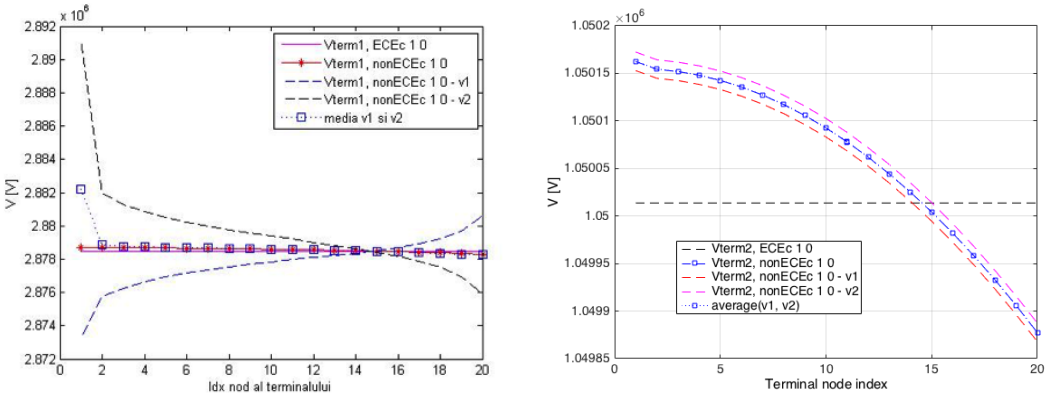


Figure 4.7: Electrical potential of the near (left) and far (right) end electrodes for various implementations of the nonECE boundary conditions.

Table 4.4: The relative errors of the models reduced with VF.

	$\varepsilon_{\text{rel}}[\%], \text{Y}_{11}$				$\varepsilon_{\text{rel}}[\%], \text{Y}_{12}$			
L	$q=1$	$q=2$	$q=3$	$q=9$	$q=1$	$q=2$	$q=3$	$q=9$
$\lambda_0/4$	0.69	$1.5 \cdot 10^{-5}$	$9.24 \cdot 10^{-8}$	$4.8 \cdot 10^{-11}$	2.77	$0.97 \cdot 10^{-5}$	$2.75 \cdot 10^{-7}$	$5.88 \cdot 10^{-13}$
λ_0	39	1.21	0.03	$5.00 \cdot 10^{-9}$	23.87	2.81	0.05	$1.02 \cdot 10^{-10}$
$2.5\lambda_0$	23.79	3.38	0.4	$1.00 \cdot 10^{-7}$	7.42	4.57	1.52	$3.86 \cdot 10^{-10}$

two alternatives 1 and 2 are bounds for the correct distribution, but the best numerical solution is obtained using the final version based on a dual network [66].

The EQS analysis requires minimal changes of the MATLAB code (available in Appendix A1), namely the cell conductances G are replaced with their admittances, which are obtained by simply replacing the membrane conductivity σ_2 with $\sigma_2 + j\omega\varepsilon$. The results now depend on the frequency, and their representation in Fig. 4.6 overlaps those obtained with FEM, which validates both numerical methods FEM and FIT, in the case of ECE boundary conditions. The FEM solution is more accurate, but the computational effort is higher, since the system has four times more equations. To overcome this drawback, the dFIT approach can be used [163], in which the unknowns are not only the potentials in nodes but also those in the center of the cells. The number of unknowns doubles, but the advantage is that two independent systems, each with n equations, are separately solved in parallel or sequentially for the two sets of potentials. Moreover, the two solutions realize a bracketing of the exact solution, allowing the control of the numerical computing error, since their difference is an upper bound of that error.

The EQS analysis needed a solving time of around 1.42 sec on a two-core MacOS system and used 1.604 MB of memory.

4.2 Order reduction of the 2.5D model

In order to obtain simpler models valid for a large range of frequencies, methods such as Vector Fitting (VF) may be used [164], [165]. VF is a data-oriented reduction method, which searches for a rational approximation of the frequency characteristic given as a set

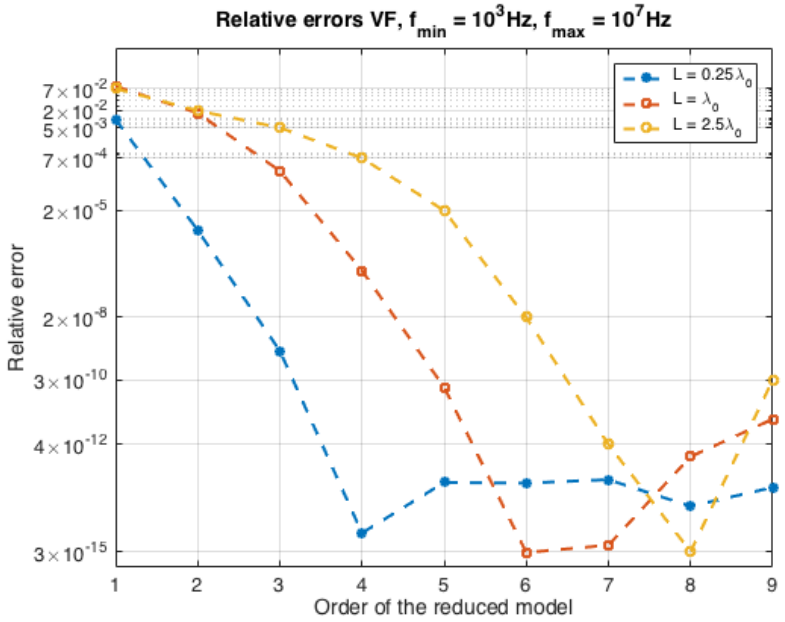


Figure 4.8: The relative errors of the reduced model (Y_{11} and Y_{12}) with VF vs. order q.

of samples.

The order of the reduced system is imposed by the user and the result represents the best approximation of that order. Starting from a set of N_s circuit matrices $\mathbf{H} = \{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_{N_s}\}$ of size $p \times p$ (in this case $p = 2$ and the matrices are positive defined and symmetrical), representing the admittances corresponding to the frequencies $s = [s_1, s_2, ..., s_{N_s}]$ with $s_k = j2\pi f_k$, it is aimed to iteratively determine the parameters of the rational matrix function in s:

$$\mathbf{Y}(s) = \mathbf{D} + s\mathbf{E} + \sum_{k=1}^{q} \frac{\mathbf{R}_k}{s - p_k}, \tag{4.50}$$

which approximates the data. It has q poles p_k, with residues $\mathbf{R}_k$ which are symmetrical complex matrices of dimension $p \times p$ and the real symmetrical matrices $\mathbf{D}$ and $\mathbf{E}$ of the same size. Using the data from the $N_s = 10$ frequency samples represented in Fig. 4.6, reduced models of orders $q = 1, 2, 3, ..., 9$ were successively retrieved. The relative errors of the reduced models for different compartment lengths are presented in Table 4.4.

Fig. 4.8 shows the relative errors of the reduced 2.5D model (with admittances Y_{11} and Y_{12}) with VF, for different line lengths and different orders q. Errors less than 10^{-10} are obtained for orders ranging from 4 to 8. For practical applications a order $q = 3 \div 5$ provides acceptable accuracy. For example, for $q = 3$ and $L = 2.5\lambda_0$, the three poles of the reduced model are $p_1 = -2.25 \cdot 10^8$, $p_2 = -2.30 \cdot 10^7$, $p_3 = -4.26 \cdot 10^6$ and the zeros are $z_1 = -2.61 \cdot 10^7$, $z_2 = -4.42 \cdot 10^6$. The real and negative values of the poles guarantee the stability and non-oscillating character of the reduced model. The conductance $G_{11} = 0.782 \cdot 10^{-6}$ of the reduced model has a deviation of 0.2% from the d.c. value of $Y_{11}(s)$.

Table 4.5: The asymptotic limits for Y_{11} and Y_{12}.

Length L	Y_{11}, Y_{12} for $f \to 0$	Y_{11}, Y_{12} for $f \to \infty$
$L \to 0$	$Y_{11} \to \infty$, $Y_{12} \to \infty$	∞, ∞
$L = \lambda_0$	$Y_{11} = \mathrm{th}(1)/Z_0$, $Y_{12} = 1/Z_0$	∞, 0
$L \to \infty$	$Y_{11} = 1/Z_0$, $Y_{12} \to 0$	$Y_{11} \to \infty$, $Y_{12} \to 0$

4.3 Models of reduced order for Cable (1D) model

For the 1D case, we generate four categories of models: analytical, analytical reduced, numerical and numerical reduced.

4.3.1 Analytical 1D model

Considering the one-dimensional model in Chapter 3 (Fig. 3.8), with length L as a two-port, excited in terminal voltages, the global dynamic admittance matrix can be extracted, after solving (3.17) with zero initial conditions [166]:

$$\mathbf{Y}(s) = \begin{bmatrix} \mathrm{Y}_{11} & \mathrm{Y}_{12} \\ \mathrm{Y}_{21} & \mathrm{Y}_{22} \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{ch}(\gamma L)}{Z_C \mathrm{sh}(\gamma L)} & -\frac{1}{Z_C \mathrm{sh}(\gamma L)} \\ -\frac{1}{Z_C \mathrm{sh}(\gamma L)} & \frac{\mathrm{ch}(\gamma L)}{Z_C \mathrm{sh}(\gamma L)} \end{bmatrix}; \tag{4.51}$$

$$\begin{bmatrix} \mathrm{I}(0,s) \\ \mathrm{I}(L,s) \end{bmatrix} = \begin{bmatrix} \mathrm{Y}_{11} & \mathrm{Y}_{12} \\ \mathrm{Y}_{21} & \mathrm{Y}_{22} \end{bmatrix} \begin{bmatrix} \mathrm{V}(0,s) - V_0/s \\ \mathrm{V}(L,s) - V_0/s \end{bmatrix}. \tag{4.52}$$

The characteristic line parameters $\gamma = \sqrt{r(g+sc)} = \sqrt{1+\tau s}/\lambda_0$ and $Z_C = \sqrt{r/(g+sc)} = 1/(g\lambda_0\sqrt{1+\tau s})$ do not depend on the line length. In the static regime ($s = 0$), $\gamma = \sqrt{rg} = 4.64 \cdot 10^3\,\mathrm{m}^{-1}$ and $Z_C = Z_0 = \sqrt{r/g} = 1/(g\lambda_0) = 1.29\,\mathrm{M\Omega}$.

The linear model described by these complex admittances has an equivalent circuit with an infinite number of capacitances and resistances. Voltage sources having e.m.f. $E = V_0$ have to be connected in series with the input ports to satisfy (3.18). Table 4.5 contains the asymptotic values of admittances vs. frequency.

The frequency characteristics of the 1D model $Y_{11}(f) = |\mathrm{Y}_{11}(j\omega)|$, $Y_{12}(f) = |\mathrm{Y}_{12}(j\omega)|$, $\omega = 2\pi f$, for two lengths: $L = \lambda_0/4$ and $L = 2.5\lambda_0$, are shown in Fig. 4.9. The dotted lines are the reference values, extracted from the numerical 2.5D FEM model. The two models give similar results over the entire frequency range.

The two circuit functions Y_{11} and especially Y_{12} determine the voltage attenuation factor of the transmitted signal, which is the system's transfer function when the output terminal has a null current:

$$\begin{aligned} &\mathrm{I}(L,s) = \mathrm{Y}_{21}(s)\mathrm{V}(0,s) + \mathrm{Y}_{22}(s)\mathrm{V}(L,s) = 0 \\ &\Longrightarrow \mathrm{A}_v(s) = \frac{\mathrm{V}(L,s)}{\mathrm{V}(0,s)} = -\frac{\mathrm{Y}_{21}(s)}{\mathrm{Y}_{22}(s)} = \frac{1}{\mathrm{ch}(\sqrt{1+s\tau}L/\lambda_0)}. \end{aligned} \tag{4.53}$$

In stationary regime, the signal attenuation is $A_0 = \mathrm{A}_v(0) = 1/\mathrm{ch}(L/\lambda_0)$. The system acts as a low-pass filter, with $\mathrm{A}_v(s) \xrightarrow{s\to\infty} 0$ and an infinity of poles s_k, all real and negative and satisfying the relation: $s_k\tau + 1 = ((k-1/2)\pi\lambda_0/L)^2$, the first being essential. The small length compartment does not attenuate the input voltage in stationary regime, Y_{11}

Table 4.6: The global relative errors of the analytical 1D model.

Length L	$\varepsilon_{rel}[\%], Y_{11}$	$\varepsilon_{rel}[\%], Y_{12}$
$L = \lambda_0/4$	0.079	0.09
$L = \lambda_0$	0.055	0.027
$L = 2.5\lambda_0$	0.057	0.003

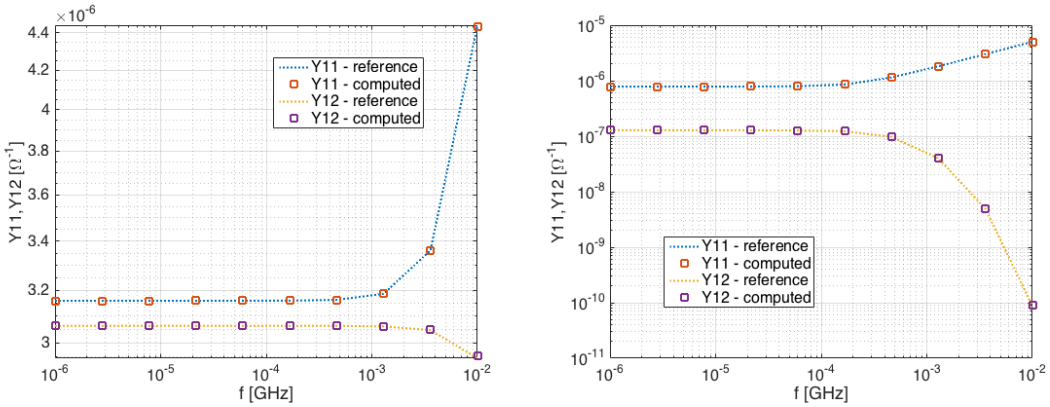


Figure 4.9: The frequency characteristics, $Y_{11}(f)$, $Y_{12}(f)$ for $L = \lambda_0/4$ (top) and $L = 2.5\lambda_0$ (bottom), computed with (4.51).

and Y_{12} being practically equal. As expected, the attenuation is higher for larger line lengths, six times higher for $L = 2.5\lambda_0$ than for short line.

Low frequency errors are very different from errors computed at high frequencies. Under these conditions, the global error estimate by the classical method gives completely irrelevant results. Therefore it is necessary to define the error differently. The typical neuronal signal has a spectrum in which the low frequency components are much more significant than the high frequency ones. In order to correctly quantify the error, a weighted norm $w(f)$, adequately defined is proposed [167]:

$$\|\mathbf{Y}\|_w \cong \int_{f_m}^{f_M} w(f)\|\mathbf{Y}(f)\|_2 \mathrm{d}f \cong \sum_{k=0}^{M} w(f_k)\|\mathbf{Y}(f_k)\|_2 \Delta f_k. \tag{4.54}$$

Here $\|\mathbf{Y}\|_2$ is the Euclidian norm of the matrix $\mathbf{Y}$ and the weight $w(f)$ is computed from the spectrum $w'(f)$ of the standard neuronal signal $s(t)$:

$$\begin{aligned} w'(f) &= \left| \int_0^{\infty} s(t) e^{-j2\pi ft} \mathrm{d}t \right| \\ &\cong \frac{t_{max}}{N} \left| \sum_{k=0}^{N} s\left(kt_{max}/N\right) e^{-j2\pi fkt_{max}/N} \right|; \\ W &= \int_0^{\infty} w'(f)\mathrm{d}f \cong \int_{f_m}^{f_M} w'(f)\mathrm{d}f \cong \sum_{k=0}^{M} w'(f_k)\Delta f_k. \end{aligned} \tag{4.55}$$

Thus the normalized weight $w(f_k) = w'(f_k)/W$ has a unitary integral. Considering the weighted norm, the global error is defined as:

$$\varepsilon_{rel} = Z_0 \left\| \mathbf{Y}_{1D} - \mathbf{Y}_{2.5D} \right\|_w, \tag{4.56}$$

where $Z_0 = 1/(g\lambda_0)$ is the d.c. characteristic impedance of the line.

Table 4.6 contains the relative errors between the 1D model and the reference 2.5D

FEM model for Y_{11} and Y_{12}, computed with (4.56) for 10 frequencies between f_m and f_M, for different line lengths.

The general conclusion is that the 1D "cable model" correctly describes the transmission of the neuronal signal through the myelinated compartment, having numerical deviations from the field model of under 0.1%, even smaller for optimal lengths of real compartments (which is larger than the characteristic length). Consequently, if a better accuracy is required, the cable model is not acceptable and 2.5D field models should be used, but for the most practical cases this accuracy is acceptable [66].

4.3.2 Order reduction of the Analytical 1D model – simple fractions, Taylor-Padé, truncated products

The hyperbolic functions which compose the analytical model can be expanded in different infinite series and products [168]:

$$\frac{1}{\mathrm{sh}(z)} = \frac{1}{\sum_{k=0}^{\infty} \frac{z^{2k+1}}{(2k+1)!}} = \frac{1}{z} + 2z\sum_{k=1}^{\infty} \frac{(-1)^k}{z^2+\alpha_k^2} = \frac{1}{z\prod_{k=1}^{\infty}\left(1+\left(\frac{z}{\alpha_k}\right)^2\right)}, \alpha_k = k\pi; \quad (4.57)$$

$$\mathrm{cth}(z) = \frac{\sum_{k=0}^{\infty} \frac{z^{2k}}{(2k)!}}{\sum_{k=0}^{\infty} \frac{z^{2k+1}}{(2k+1)!}} = \frac{1}{z} + 2z\sum_{k=1}^{\infty} \frac{1}{z^2+\alpha_k^2} = \frac{\prod_{k=1}^{\infty}\left(1+\left(\frac{z}{\beta_k}\right)^2\right)}{z\prod_{k=1}^{\infty}\left(1+\left(\frac{z}{\alpha_k}\right)^2\right)}, \beta_k = (2k-1)\pi/2. \quad (4.58)$$

The first way of expressing the hyperbolic functions in (4.57) and (4.58) is a series of powers (Taylor-Padé for the numerator and denominator), the second is a series of simple fractions, and the last is a product of poles and zeros. Consequently, this model has an infinite divergent sequence of poles, distributed in an arithmetic progression on the negative semi-axis, the zeros being intercalated in-between the poles. We should notice that α_k are exactly the proper values of the spatial operator of the transmission equation, so they represent the modal characteristics of this equation.

Different reduced models – characterized by rational functions – are obtained by truncating these series to q terms. The first q poles of these functions have the smallest absolute value and they represent the most relevant poles of the original function. The truncation of the power series modifies the position of the poles, possibly generating complex poles, as it happens very early in the test case, for $q = 3$. The big advantage of these reducing approaches is that the reduced model is natively parametric. For example, reducing with simple fractions gives:

$$Y_{11}(s) = \frac{\mathrm{cth}(\gamma L)}{Z_C} \cong \frac{g\lambda_0^2}{L}\left(1 + 2\sum_{k=1}^{q}\frac{s\tau+1}{s\tau+1+(\alpha_k\lambda_0/L)^2}\right).$$

Unfortunately, the truncation in simple fractions of Y_{12} has a very slow convergence; therefore it is preferable to use Taylor-Padé or infinite product truncation for this circuit function. Besides the methods presented above, there is also the possibility to use continuous fractions expansion (authentic Padé) [169], which ensures moments' conservation and is expected to have the fastest convergence. In our test case, this method gave best

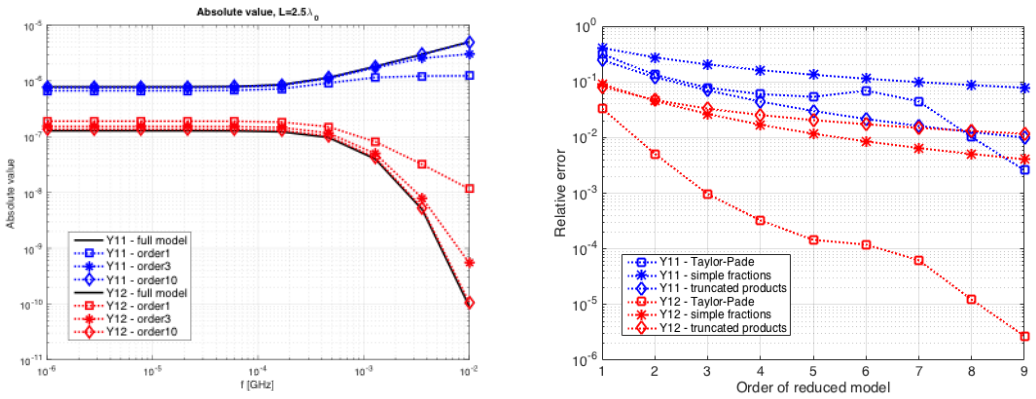


Figure 4.10: Left: Frequency characteristics of the reduced 1D model of different orders, at $L = 2.5\lambda_0$; the reference values, extracted from the analytical 1D model are represented with a black continuous line. Right: relative error of the 1D cable model reduced by series truncation vs. order of the reduced model, $L = 2.5\lambda_0$.

results (minimal errors) for order 10, but for order 3 the errors were between Taylor-Padé and simple fractions.

For the line lengths of interest, truncating the series of powers and truncating the infinite products proved to be the best reducing methods. The latter may be because truncating the infinite products takes into account both poles and zeros of original transfer functions. It balances both perspectives: current and voltage excitation. Fig. 4.10 shows the frequency characteristics of the reduced model and the way the relative error varies with the order, for different methods and $L = 2.5\lambda_0$.

Table 4.7 shows the relative errors of the reduced models (Y_{11} and Y_{12}, obtained by simple fractions, Taylor-Padé truncation and product truncation) for different line lengths and different orders (q), considering the analytical 1D model as reference.

With simple fractions reduction, Y_{12} has small relative errors, of less than 3% for large line and order 3. However, the deviations of Y_{11} for lengths of interest such as $2.5\lambda_0$ are much larger, of about 40% for order 1 and 20% for order 3. With Taylor-Padé reduction, both Y_{11} and Y_{12} have small relative errors, of 8% and 0.1% respectively, for large line and order 3. However, the deviation of Y_{11} for lengths of interest – such as $2.5\lambda_0$ – at order 1 is still quite large, of 31%. With product truncation reduction, Y_{11} has the smallest deviation at order 1, of less than 25%. For larger orders, the deviations of both Y_{11} and Y_{12} are similar and relatively small, but larger than the relative errors obtained with Taylor-Padé.

Therefore, the Taylor-Padé method gives the best approximation:

$$Y_{11}(s) = g\lambda_0 \frac{1 + \sum_{k=1}^{q} \frac{(L/\lambda_0)^{2k}(1+s\tau)^k}{(2k)!}}{L/\lambda_0 + \sum_{k=1}^{q} \frac{(L/\lambda_0)^{2k+1}(1+s\tau)^k}{(2k+1)!}};$$

$$Y_{12}(s) = g\lambda_0 \frac{1}{L/\lambda_0 + \sum_{k=1}^{q} \frac{(L/\lambda_0)^{2k+1}(1+s\tau)^k}{(2k+1)!}},$$

excepting Y_{11}, for order 4-7 where truncated products give better results.

For example, for $q = 4$ and $L = 2.5\lambda_0$, the four poles of the reduced model with

Table 4.7: The relative errors of the reduced analytical 1D model vs. its order, Y_{11} and Y_{12}.

Length L	$\varepsilon_{rel}[\%]$ $q=1$	$\varepsilon_{rel}[\%]$ $q=3$	$\varepsilon_{rel}[\%]$ $q=10$
Y_{11}			
$\lambda_0/4$, simple fractions	4.81	2.11	0.71
λ_0, simple fractions	18.83	8.45	2.84
$2.5\lambda_0$, simple fractions	40.61	20.52	7.09
$\lambda_0/4$, Taylor-Padé	0.36	1.68e-4	1.2e-13
λ_0, Taylor-Padé	9.13	1.04	8.28e-9
$2.5\lambda_0$, Taylor-Padé	31.62	7.85	0.05
$\lambda_0/4$, truncated products	1.11	0.18	0.018
λ_0, truncated products	7.82	1.39	0.14
$2.5\lambda_0$,truncated products	24.57	7.00	0.81
Y_{12}			
$\lambda_0/4$, simple fractions	1.32	0.28	0.03
λ_0, simple fractions	5.04	1.14	0.13
$2.5\lambda_0$, simple fractions	9.11	2.64	0.33
$\lambda_0/4$, Taylor-Padé	0.09	2.08e-5	1.4e-13
λ_0, Taylor-Padé	1.71	0.073	1.85e-10
$2.5\lambda_0$, Taylor-Padé	3.34	0.096	5.18e-5
$\lambda_0/4$, truncated products	2.38	1.04	0.35
λ_0, truncated products	7.35	3.19	1.06
$2.5\lambda_0$,truncated products	8.15	3.30	1.05

Table 4.8: The global relative errors of the numerical 1D model for $L = 2.5\lambda_0$, with respect to the analytical 1D model.

$\varepsilon_{rel}[\%]$	$q=1$	$q=3$	$q=10$	$q=100$
Y_{11}	73.39	43.17	16.68	1.83
Y_{12}	24.83	2.75	0.25	1.96e-03

Table 4.9: The local relative errors of the numerical 1D model at f_m and f_M.

L	$\varepsilon_{rel}[\%], q$=3		$\varepsilon_{rel}[\%], q$=10		$\varepsilon_{rel}[\%], q$=100	
	at f_m	at f_M	at f_m	at f_M	at f_m	at f_M
$\lambda_0/4$	25.1	24.7	9.13	8.63	0.99	0.93
λ_0	26.8	66.7	9.75	27.8	1.06	3.1
$2.5\lambda_0$	32.5	81.5	11.5	61.8	1.22	7.78

truncated products are $p_1 = -3.85 \cdot 10^6$, $p_2 = -1.09 \cdot 10^7$, $p_3 = -2.27 \cdot 10^7$, $p_4 = -3.92 \cdot 10^7$ and the zeros are $z_1 = -2.08 \cdot 10^6$, $z_2 = -6.79 \cdot 10^6$, $z_3 = -1.62 \cdot 10^7$, $z_4 = -3.03 \cdot 10^7$, whereas G_{11} has a deviation of 0.35% from the d.c. value of admittance $\mathrm{Y}_{11}(s)$.

4.3.3 Numerical 1D model: reduction by segmentation

As stated in Chapter 3, solving the equation (3.18) numerically implies the discretization of the interval $0 < x < L$ in a grid with q nodes (plus the peripheral ones, indexed 0 and $q+1$), which are assumed to be spaced with the step $h = \Delta x$. By finite centered differences, a system of q ODEs is obtained:

$$\frac{V_{k-1}(t) - 2V_k(t) + V_{k+1}(t)}{\Delta x^2} = \frac{1}{\lambda_0^2}\left(V_k(t) - V_0\right) + \frac{\tau}{\lambda_0^2}\frac{\mathrm{d}V_k(t)}{\mathrm{d}t}$$

$$k = 1, .., q;\ \frac{\Delta x^2}{\lambda_0^2} = rg\Delta x^2 = RG,\ \Delta x^2\frac{\tau}{\lambda_0^2} = rc\Delta x^2 = RC,$$

$$\text{with } R = r\Delta x, G = g\Delta x, C = c\Delta x,$$

which after Laplace transform becomes:

$$-\frac{1}{R}V_{k-1}(t) + \left(\frac{2}{R} + G + sC\right)V_k(t) - \frac{1}{R}V_{k+1}(t) = GV_0.$$

These are actually the nodal equations of the circuit in Fig. 3.21, with $V_0 = 0$ and we call this approach "reduction by segmentation". This circuit has q capacitors, so it is a linear circuit with q state variables.

Having null internal sources ($V_0 = 0$) and controlling the terminals with $V_k = E = 1V$, for $k = 0$ and $V_{q+1} = 0$, the admittances are extracted by solving the linear system of nodal equations, from which only the first and the last potential are of interest: $Y_{11} = Y_{22} = (E - V_1)/R$ and $Y_{12} = Y_{21} = V_q/R$, numerically equal with the terminal currents. Table 4.8 shows the relative errors computed with (4.56) of the numerical models with 1, 3, 10, 100 segments (q), considering the analytical model (4.51) as reference, in the case of $L = 2.5\lambda_0$.

Table 4.9 shows the relative errors of the numerical model at f_m and f_M for different line lengths and different orders (q), considering the analytical 1D model as reference.

Second order finite centered differences have the order of error [154]:

$$|E_k| = |V_k - U_k| \le \frac{(\lambda_0 h)^2}{12} = \frac{(L/\lambda_0)^2}{12q^2},$$

$$c_4 = \max_{0<x<L}\left(\left|\frac{\partial^4 V}{\partial x^4}\right|\right) \le \frac{1}{\lambda_0^4},$$

quadratic to the discretization step. It is therefore expected that as q increases, the error decays inversely proportional to its square.

4.3.4 Order reduction of Numerical 1D model – BT, Krylov, POD

The numerical solving of the TL equation implicitly reduces the order, from infinite (the system order before meshing) to a finite number q. This technique of reducing the order is very widespread in the literature, and there are recommendations for choosing

the value $5 \div 10$ for q [92]. We are not aware of any credible study referring the error induced by this choice. However it is clear that the reduction by segmentation does not necessarily lead to an optimal result for a given order. We aim to find that smaller errors can be obtained for the same order if we reduce the state system of the circuit with n segments to a system with $q \ll n$ state variables. The equivalent circuit with refined segmentation has the state equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} V_1 \\ .. \\ .. \\ .. \\ V_n \end{bmatrix} = \begin{bmatrix} .. & .. & .. & .. & .. \\ .. & -2-RG & 1 & 0 & .. \\ .. & 1 & -2-RG & 1 & .. \\ .. & 0 & 1 & -2-RG & .. \\ .. & .. & .. & .. & .. \end{bmatrix} \begin{bmatrix} V_1 \\ .. \\ .. \\ .. \\ V_q \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ .. \\ .. \\ 0 \end{bmatrix} \frac{E}{R}$$

$$\Leftrightarrow \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}v + \mathbf{B}e; \mathbf{v} \in \mathbb{R}^{n\times 1}, \mathbf{A} \in \mathbb{R}^{n\times n}; \mathbf{B} \in \mathbb{R}^{n\times 1}.$$

The potentials V_1 and V_n are output signals, whereas E is the input signal of the system. This state system may be reduced by different model order reduction techniques, such as [170], [22], [171]:

1. **Balanced truncation**

   From the MOR methods, balanced truncation (BT) has the most solid theoretical foundation [172]. It is a projection method, based on the calculation of observability and controllability Gramian matrices of the system. Consequently, the cost of this method is relatively high, but the result is of optimal quality, ensuring accuracy control for the reduced model.

   If the system is stable, controllable and observable, then the Gramian observability and controllability matrices can be computed: $\mathbf{W}_c = \mathbf{W}_o = diag(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_n \leq 0$ as solutions of the Lyapunov equations:

   $$\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^{\mathrm{T}} = -\mathbf{B}\mathbf{B}^{\mathrm{T}};$$
   $$\mathbf{A}^{\mathrm{T}}\mathbf{W}_o + \mathbf{W}_o\mathbf{A} = -\mathbf{B}^{\mathrm{T}}\mathbf{C}.$$

   The spectrum of the matrix $\mathbf{W}_c\mathbf{W}_o$ is given by the Hankel singular of the system: $\sqrt{\lambda(\mathbf{W}_c\mathbf{W}_o)} = \{\sigma_1, \ldots, \sigma_n\}$.

   An arbitrary system is transformed into a balanced one $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{C}})$ through the transformation $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{C}}) = (\mathbf{T}\mathbf{A}\mathbf{T}^{-1}, \mathbf{T}\mathbf{B}, \mathbf{C}\mathbf{T}^{-1})$, which has the Gramian matrices $\mathbf{W}_c = \mathbf{T}\widetilde{\mathbf{W}}_c\mathbf{T}^{\mathrm{T}} = \mathbf{W}_o = \mathbf{T}^{-\mathrm{T}}\widetilde{\mathbf{W}}_o\mathbf{T}^{-1}$.

   The balanced systems are descending ordered internally into parts les and less observable and controllable:

   $$(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{C}}) = \left(\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}, \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix}, \begin{bmatrix} \tilde{C}_1 & \tilde{C}_2 \end{bmatrix} \right)$$

   After eliminating the least important part the truncated reduced system results: $(\widetilde{\Sigma} = \widetilde{\mathbf{A}}_{11}, \widetilde{\mathbf{B}}_1, \widetilde{\mathbf{C}}_1)$. The projection matrix is obtained by truncating the transformation matrix used for balancing. The truncation error for a reduced system of order r is bounded by:

   $$\|\Sigma - \widetilde{\Sigma}\|_2 \leq 2\|u\|_2 \sum_{k=r+1}^{n} \sigma_k \tag{4.59}$$

both in Euclidean and Cebisev norm.

The transformation for balancing can also be determined by singular value decomposition technique [22]. MATLAB has several functions for order reduction by balancing [173], [174], [175]. Suplimentary information on this reduction method can be found in [176] (for the interpretation of Gramians) and [177] (for history and alternatives).

The high complexity of this approach makes it impossible to apply to large-scale systems. An approximate low cost approach that links the balanced truncation (TBR) approximations and multipoint rational approximation approaches in frequency domain is Poor man's TBR [178].

In our case, the Bode diagrams of the 100-segments system and the characteristics of the reduced systems using balanced truncation to orders from 1 to 10 tend to close in as the q order rises. The poles and zeros of the reduced model of order 3 have the values: $z_1 = -405.54 \cdot 10^6$, $p_1 = -69.66 \cdot 10^6$; $z_2 = -10.13 \cdot 10^6$, $p_2 = -9.66 \cdot 10^6$; $p_3 = -2.21 \cdot 10^6$, real, negative and alternate, and $G_{11} = 0.812 \cdot 10^{-6}$ has a deviation of 4.91% from the d.c. value of admittance $\mathbf{Y}(s)$.

The model error drops below 1% only for orders $q > 6$, whereas for $q = 3$ it is about 10%, values about three times smaller than for the model obtained by simple segmentation.

2. **Krylov Subspace (projection) methods**

In this class of methods, the state equations of large systems are reduced by projection on a Krylov subspace. They are considered to be among the best methods for order reduction from a quality – cost (computing effort) ratio point of view [179], [180].

For a square matrix $\mathbf{A}$ the Krylov subspace is the linear space generated by the basis vectors:

$$\mathrm{K_q}(\mathbf{A}, \mathbf{r}) = \mathrm{span}\left\{\mathbf{r}, \mathbf{A}\mathbf{r}, \mathbf{A}^2\mathbf{r}, \ldots, \mathbf{A}^{\mathrm{q}-1}\mathbf{r}\right\},$$

where $\mathbf{r}$ is the first basis vector. A projection on this subspace of dimension q is a polynomial in matrix $\mathbf{A}$. If $\mathbf{A}$ is not symmetrical the left Krylov subspace can be defined, having $\mathbf{A}^{\mathrm{T}}$ replacing $\mathbf{A}$.

The Krylov subspaces are important, since their coordinates are related to the coefficients of the Taylor series of the transfer function, expanded around the point $s = 0$ (they are called the system's moments, being matrices in the MIMO case and vectors in the SISO case):

$$\mathbf{M}_i = \mathbf{C}\mathbf{A}^{-(i+1)}\mathbf{B},\ i = 0, 1, \ldots.$$

The moments can be defined around a different point of expansion s_0 other than the origin, including infinity (in this cased they are called Markov parameters). The Krylov space has a much wider importance, in the iterative solving of linear algebraic systems, since the solution of the system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ lies in the Krylov space $K_q(\mathbf{A}, \mathbf{b})$ of dimension equal to the degree of the minimal polynomial P of matrix $\mathbf{A}$, for which $P(\mathbf{A}) = 0$ [181].

The reduced order model is obtained through the projection realized by the linear transformation $\mathbf{V}$, which reduces the state vector size from n to q:

$$\mathbf{x} = \mathbf{V}\mathbf{x}_\mathrm{r}.$$

Multiplying the state equations with the transpose of $\mathbf{W}$ (of size $q \times n$, often equal to $\mathbf{V}$), having $\mathbf{W}^\mathrm{T}\mathbf{V} = \mathbf{I}$, leads to the reduced system of order q:

$$\begin{cases} \mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u} \\ \mathbf{y} = \mathbf{c}^\mathrm{T}\mathbf{x} \end{cases},$$

$$\mathbf{x} = \mathbf{V}\mathbf{x}_\mathrm{r} \Longrightarrow \begin{cases} \mathbf{V}\dot{\mathbf{x}}_\mathrm{r} = \mathbf{A}\mathbf{V}\mathbf{x}_\mathrm{r} + \mathbf{b}\mathbf{u} \\ \mathbf{y} = \mathbf{c}^\mathrm{T}\mathbf{V}\mathbf{x}_\mathrm{r} \end{cases} \Longrightarrow \begin{cases} \dot{\mathbf{x}}_\mathrm{r} = \mathbf{A}_\mathrm{r}\mathbf{x}_\mathrm{r} + \mathbf{b}_\mathrm{r}\mathbf{u} \\ \mathbf{y} = \mathbf{c}_\mathrm{r}^\mathrm{T}\mathbf{x}_\mathrm{r} \end{cases},$$

with $\mathbf{A}_\mathrm{r} = \mathbf{W}^\mathrm{T}\mathbf{A}\mathbf{V}, \mathbf{b}_\mathrm{r} = \mathbf{W}^\mathrm{T}\mathbf{b}, \mathbf{c}_\mathrm{r}^\mathrm{T} = \mathbf{c}^\mathrm{T}\mathbf{V}$.

The fundamental theorem at the basis of this reduction procedure states that if $\mathbf{V}$ is the basis of the Krylov space $K_q\left(\mathbf{A}^{-1}, \mathbf{A}^{-1}\mathbf{b}\right)$ and $\mathbf{A}_\mathrm{r}$ is non-singular, then the original and the reduced system have the first q moments equal.

By conveniently choosing the matrix $\mathbf{W}$ and the start vector, it is possible to have the equality of more than q characteristic parameters (Markov moments or parameters) between the original and the reduced system. For example, if $\mathbf{W}$ is a base of a left Krylov space $K_q\left(\mathbf{A}^{-\mathrm{T}}, \mathbf{A}^{-\mathrm{T}}\mathbf{c}\right)$, then the two systems have the first $2q$ moments equal in origin and we say that we have applied a bilateral procedure.

In most cases, the basis vectors defining the Krylov subspace tend to become almost linearly dependent even for small values of q, which is why they cannot be used as such in numerical calculations. Robust (stable and computationally efficient) algorithms of order reduction use procedures to generate a base as close as possible to an orthogonal one (Lanczos and Arnoldi procedures). In the Arnoldi algorithm, the basis vectors are orthogonal, which ensures good accuracy. Another popular algorithm is Lanczos, which is a bilateral procedure with two orthogonal Krylov spaces, thus ensuring $2q$ equal moments. A detailed description of these algorithms can be found in several documents, including [170].

If the algorithm also ensures the equality of Markov parameters (the expansion is done at infinity), then we say that we have made a Padé approximation, and if the expansion point is an arbitrary point in the complex plane s_0, then we say we have a "rational interpolation", rational Lanczos / Arnoldi procedures providing efficient solutions with good accuracy around the frequency of interest. Among the algorithms in this category we mention three: PvL ("Padé via Lanczos") presented in [182], "multipoint rational interpolation" [183] and "implicitly restarted dual Arnoldi" [184].

PvL exploits the close connection between the Lanczos process and the moment fitting technique, being very successful in the rapid simulation of large electrical circuits. The multipoint rational interpolation ensures that the transfer function moments are in a selected frequency range, providing a good approximation for a wider range of frequencies, even with low-order functions. These approaches also have limitations, such as the absence of rigorous error bounds and the non-automatic choice of sampling frequencies by the user. The Arnoldi dual approach seeks to

achieve a reduction performance similar to that obtained by balanced truncation, using bilateral Krylov spaces both for input/controllability and for output/observability. The reduction methods based on Krylov spaces have to be chosen with care to ensure the desired accuracy in the frequency of interest for each application; this is not necessary in the case of the balanced truncation, which for this reason is said to have a global character. But obviously this advantage also has a considerable cost. This is the reason why several techniques have been proposed that attempt to combine the advantages of both approaches: TBR and Krylov. More details on reduction algorithms based on projection on Krylov subspaces are available in [185], [186], [187].

The reduction procedure used on our case is RKFUN [180], an algorithm based on Rational Krylov.

The model reduced with this method from the state system with 100 degrees of freedom to $q = 2, 4, 6, 8, 10$ has the frequency characteristics practically overlayed with the full model, starting from $q = 4$. The errors are comparable with those obtained with the BT method, slightly smaller for $q = 4$ but larger for $q = 10$. In particular, the poles and zeros of the 4^{th} order reduced system: $p_1 = -3055.81 \cdot 10^6$; $p_2 = -73.27 \cdot 10^6$; $p_3 = -12 \cdot 10^6$; $p_4 = -3.85 \cdot 10^6$; $z_1 = -5119.28 \cdot 10^6$; $z_2 = -357.91 \cdot 10^6$; $z_3 = -4.57 \cdot 10^6$ show the stable character of this model, and $G_{11} = 0.781 \cdot 10^{-6}$ is deviated by 0.85% from the d.c. value of admittance $\mathbf{Y}(s)$.

3. **Proper Orthogonal Decomposition (POD)**

POD is a MOR method based on processing input and solution samples at different time moments. As opposed to the methods analyzed before, which operate on the system's equations and its structure, POD is data-oriented. This gives POD the extra advantage of being suitable to nonlinear systems as well. It is a relatively superficial approach that does not analyze the system in depth, but only some of its characteristics, reflected in its behavior towards a particular input signal. However, this is what makes POD easy to use in many situations at a low cost, even though not always with the best accuracy. It is known that in the case of linear systems, knowing the response to some particular input signals is sufficient to fully characterize the system.

Consider the nonlinear system described by:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), u(t)) \tag{4.60}$$

$$\text{and } \mathbf{X} = [\mathbf{x}(t_1), \mathbf{x}(t_2), ..., \mathbf{x}(t_N)] \in \mathbb{R}^{M \times N} \tag{4.61}$$

a collection of solution samples $\mathbf{x} \in \mathbb{R}^M$ (the state variables at different moments in time t_j, $j = 1, \ldots, N$). The samples matrix is decomposed in singular values (SVD):

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \tag{4.62}$$

where the matrices $\mathbf{U}$, $\mathbf{U}\mathbf{U}^* = \mathbf{I}_M$ and $\mathbf{V}$, $\mathbf{V}\mathbf{V}^* = \mathbf{I}_N$ are orthogonal, their columns being the left and right singular vectors of $\mathbf{X}$ and $\mathbf{\Sigma}$ is a diagonal matrix of the singular values of $\mathbf{X}$.

The singular values give information on the linearly independent character of matrix

$\mathbf{X}$, thus implicitly on the rank of the matrix. If $\sigma_r > 0$ and $\sigma_{r+1} = 0$, then the rank of matrix $\mathbf{X}$ is r. The SVD factorization expresses $\mathbf{X}$ as a sum of dyadic products $\mathbf{X} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \ldots + \sigma_r \mathbf{u}_r \mathbf{v}_r^*$.

The decomposition is then truncated by keeping only the first $k \ll r$ most important singular values. SVD allows the identification and elimination of the "almost singular" part of the matrix, that is the lines that are almost linearly dependent and keeping the best conditioned part. The low-rank approximation matrix $\widetilde{\mathbf{X}} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^*$ has the same size as $\mathbf{X}$ and the rank k.

Moreover, the deviation norm between the initial matrix and the truncated matrix satisfies the inequality:

$$\sigma_{k+1}(\mathbf{X}) < \left\| \mathbf{X} - \widetilde{\mathbf{X}} \right\|_2 < \sigma_k(\mathbf{X}) \tag{4.63}$$

In this manner a subspace is identified from the state space where the solution's time dependency path is located. The other directions of the state space are basically linear combinations of elements in the selected subspace, so they can be approximated with directions from this space. More details on the order reduction algorithms with POD are found in [188], [189].

We used an in-house POD code based on Matlab's singular value decomposition method [190].

A standard neuronal signal was used as training excitation and the snapshots matrix contains solution values in $M = 100$ spatial nodes for $N = 100$ time samples.

The snapshots were generated with an in-house program based on BTCS integration (implicit backward finite difference in time, centered differences in space). The responses of the full system and the reduced one virtually overlap from order 3. For Y_{11}, the approximation is quite accurate, whereas for Y_{12}, the characteristic of the full system is concave, whereas for the reduced system is convex. The reduced system of order 3 has the poles: $p_1 = -38.12 \cdot 10^6$; $p_2 = -4.14 \cdot 10^6$; $p_3 = -1.49 \cdot 10^6$, with much smaller absolute values than in previous cases, indicating that this model is slower. The conductance $G_{11} = 0.796 \cdot 10^{-6}$ has a deviation of 2.82% compared to the d.c. value of admittance $\mathbf{Y}(s)$. This method's error is higher than for the previous methods.

By comparing the results obtained with the three reduction methods, we conclude that the most efficient method for the studied case is the BT method.

4.4 Equivalent circuits for 0D models

The simplest model of an axonal compartment is based on the lumped parameters:

$$R = L/(\sigma_1 \pi a^2); \tag{4.64}$$

$$C = L\varepsilon_2 2\pi/\ln(b/a); \tag{4.65}$$

$$G = L\sigma_2 2\pi/\ln(b/a), \tag{4.66}$$

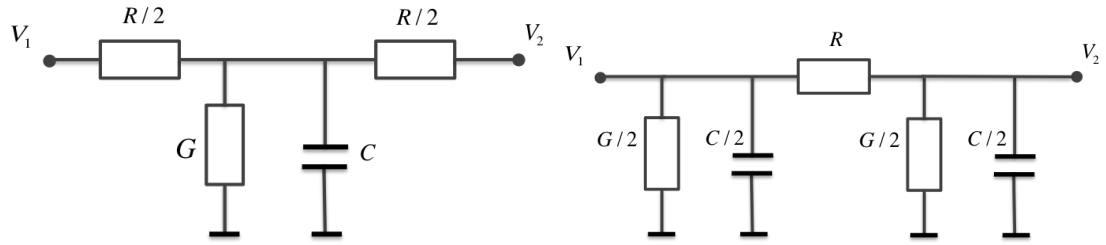


Figure 4.11: The equivalent circuits of the 0D model; left: T-type circuit; right: Π-type circuit.

Table 4.10: The admittance relative errors of 0D minimal order models.

	$\varepsilon_{\mathrm{T}}[\%]$				$\varepsilon_{\Pi}[\%]$			
	local		global		local		global	
L	at f_m	at f_M	Y_{11}	Y_{12}	at f_m	at f_M	Y_{11}	Y_{12}
$\lambda_0/4$	0.5	24.7	3.3	2.9	1.0	38.4	6.2	6.0
λ_0	7.5	81.1	19.8	6.1	15.5	271	33.5	19.2
$2.5\lambda_0$	32.1	91.5	48.7	1.2	73.9	748	114	24.8

extracted from uniform EC field in the cytoplasm and ES+EC fields in the axisymmetric membrane.

The equivalent circuit with these parameters has two possible symmetric topologies (Fig. 4.11): T-type and Π-type circuits, with the admittances matrices:

$$\mathbf{Y}_{\mathrm{T}} = \begin{bmatrix} \frac{2R(G+sC)+4}{R^2(G+sC)+4R} & -\frac{4}{R^2(G+sC)+4R} \\ -\frac{4}{R^2(G+sC)+4R} & \frac{2R(G+sC)+4}{R^2(G+sC)+4R} \end{bmatrix};$$

$$\mathbf{Y}_{\Pi} = \begin{bmatrix} \frac{RG+sRC+2}{2R} & -\frac{1}{R} \\ -\frac{1}{R} & \frac{RG+sRC+2}{2R} \end{bmatrix}.$$

The advantages of these models lie in their simplicity and in the fact that they are inherently parametric. Within the accuracy of the calculation, these models are exact; therefore their analytical character make their numerical models alike. Even though the 0D model has large errors, it is frequently used in practice as reduced order model, due to its simplicity, even though its error is often not mentioned [191], [192]. It is the model of order 1 with the smallest modeling error.

The relative errors of these two models, for 3 values of compartment lengths and 2 frequencies are given in Table 4.10, which contains also the global error for the two components Y_{11} and Y_{12}. The lumped parameter models with simpler Γ or ⅂ structures:

$$\mathbf{Y}_{\Gamma} = \begin{bmatrix} \frac{RG+sRC+1}{R} & -\frac{1}{R} \\ -\frac{1}{R} & \frac{1}{R} \end{bmatrix};$$

$$\mathbf{Y}_{⅂} = \begin{bmatrix} \frac{1}{R} & -\frac{1}{R} \\ \frac{1}{R} & \frac{RG+sRC+1}{R} \end{bmatrix}.$$

have higher global errors for $L = 2.5\lambda_0$, of 298% for Y_{11} and 24.8% for Y_{12} (Γ) and of 73.3% for Y_{11} and 24.8% for Y_{12} (⅂).

The data contained in Table 4.10 shows that this model has increasingly larger errors as the compartment length grows or as the frequency grows. The T-scheme behaves better

Table 4.11: Values of relative method errors for different models (analytical: the reference is extrapolated 2.5D FEM; numerical: the reference is the analytical model; reduced: the reference is the numerical model) [66].

Size	Analytical	Numerical	Reduced
2.5D	0.026%	$7.5 \cdot 10^{-6}$%	VF: 0.5% for $q = 3$; $3 \cdot 10^{-8}$% for $q = 9$
1D	0.057%	43% for $q = 3$; 16% for $q = 10$	Truncated products: 7% for $q = 3$; 0.8% for $q = 10$ BT: 10% for $q = 3$; 0.1% for $q = 10$
0D		48.7% (for T-scheme)	

for longer compartments, which are of practical interest ($2.5\lambda_0$ is the typical compartment length [125]), having a global error of 48.7% for Y_{11} and 1.2% for Y_{12}. If a better accuracy is required, higher order models, extracted by order reduction of 1D models should be used.

4.5 Conclusion regarding the Reduced Order Models of Myelinated Axonal Compartments

In this chapter low-order models of myelinated compartments were identified. A hierarchical series of models has been developed, corresponding to three spatial geometry classes (2.5D, 1D and 0D) and three computational approaches for each geometry (analytical, numerical and reduced order models). These models are hierarchized based on modeling errors, which are closely related to the complexity of the models.

The relative method errors for different types of models of the test problem are summarized in Table 4.11 (for $L = 2.5\lambda_0$). The simplest model is the circuit with lumped parameters (0D), but this model is not very accurate. The model with distributed parameters (cable model – 1D) is more accurate and was analyzed analytically by Laplace transformation from time domain to frequency and numerically with FDM for spatial variation. The order reduction was performed with several methods, such as: BT, projection on Krylov subspaces, POD and truncation of Taylor series or transfer function products. The most accurate model, the EQS axisymmetric (2.5D) was studied analytically using the method of variables separation (which leads to modal decomposition) and numerically with FEM, FIT and BEM methods (the most common methods used to solve PDEs). In this case, a data-driven approach (VF), was applied for order reduction.

The numerical studies conducted show that the cable model (1D) is sufficiently accurate for practical requirements and it is not worth the effort to use 2.5D models in simulation. In the mathematical literature many methods of order reduction are presented and studied, but according to our knowledge, there is no systematic study referring to their hierarchy in neuroscience applications, leading to recommendations on which is the most appropriate method in a given context, such as the one studied here.

Fig. 4.12 shows how the relative method errors depend on the order of the reduced system, for different techniques of order reduction applied to various models. The conclusion

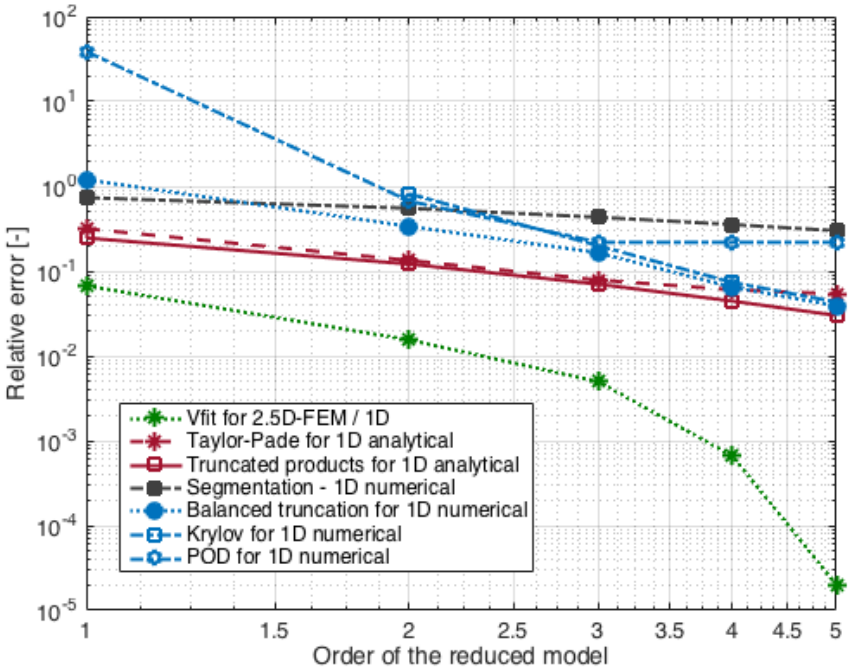


Figure 4.12: Relative error vs. order of the reduced model [66].

is that the most efficient reduction is done by the VF method, for which even for the first order the error is around 5%, and for orders 3 and 4 the error decays to 1% and to 0.1% respectively, which is satisfactory in the vast majority of practical cases. The next best method is the truncation (products and Taylor-Padé) of the transfer function (Fig. 4.12). The segmentation, which is currently the standard technique, provides results worse with at least two orders of magnitude.

Finding the best low order model for myelinated compartments is an essential step in the modeling of neuronal signals transmission along myelinated axons through saltatory conduction. The 2.5D-FEM model is the most accurate, but the 1D cable model (in analytical form) is the most computationally efficient. Since the 1D and 2.5D responses are almost identical, for the modeling of the myelinated compartments we recommend the use of the analytical 1D model (4.51), followed by reduction to order $3 \div 4$, with vector fitting (VF).

Chapter 5

Saltatory Conduction in Neurons

This chapter deals with the reduced modeling of saltatory conduction in neurons. We propose a procedure for the extraction of a reduced axon model, where reduced models of myelinated compartments are connected with nonlinear models of Ranvier nodes and the global model thus obtained is again reduced. This chapter is the development of the ideas presented in [193].

A myelinated axon has alternating sequencies of myelin covered compartments and Ranvier nodes (Fig. 5.1). The myelin sheath surrounding the axon made of glial cells works like an insulating layer, increasing the transmission speed of the action potential along the axon and reducing the energy loss across the membrane. However, the diffusion of potential in this section decreases its magnitude at the far end. If the axon were long enough, the signal at the end would not be strong enough to reach the threshold and to trigger an action potential in the next myelinated compartment. The Ranvier nodes are evenly spaced gaps in the myelin sheath, therefore uninsulated and highly rich in ion channels, allowing the exchange of ions required to regenerate the action potential.

This optimal design allows the axons to be no matter how long, provided that the myelinated compartments' length is not greater than the maximum transmission length [125], so that the potential is eligible for regeneration when entering a Ranvier node

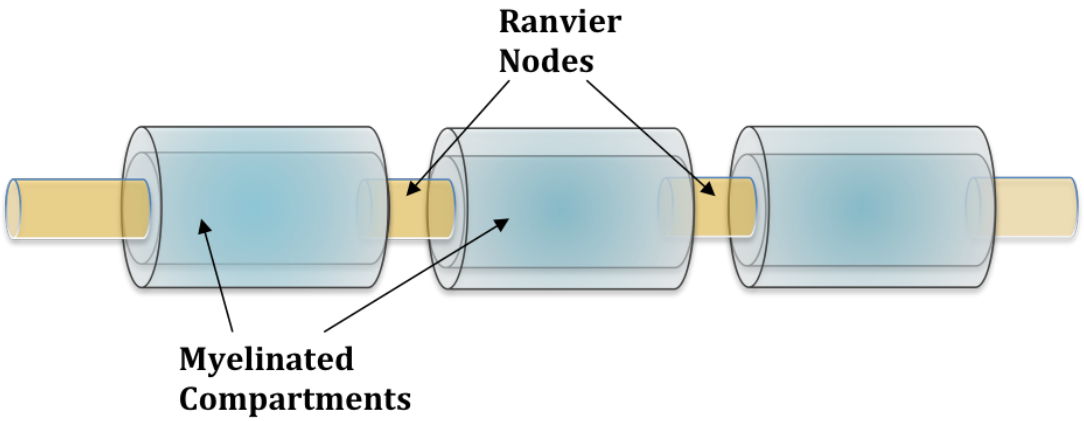


Figure 5.1: Simplified geometrical model of a myelinated axon, consisting of a chain of myelinated compartments and Ranvier nodes [193].

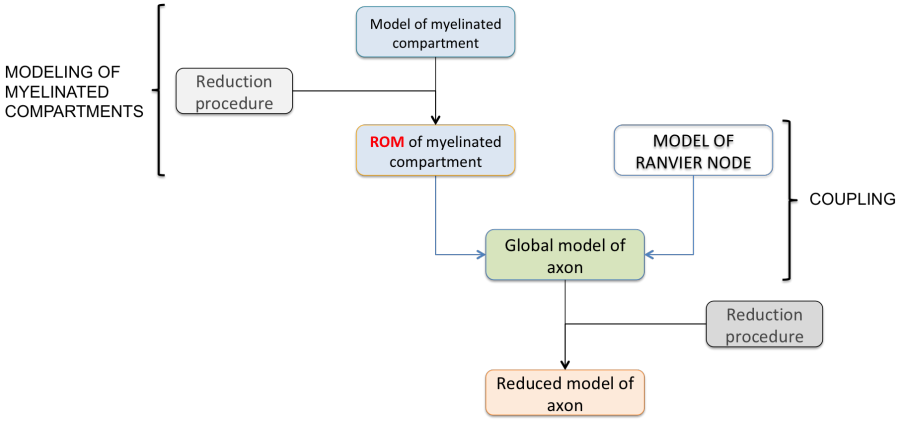


Figure 5.2: The procedure proposed for the extraction of a reduced axon model: reduced models of myelinated compartments are connected with models of Ranvier nodes and the global model thus obtained is again reduced.

(that is, the magnitude is above the threshold). The transmission of action potential in myelinated axons is called "saltatory conduction" (from the Latin word *saltare*, which means to hop), because the potential seems to jump from one Ranvier node to another.

Since it has been experimentally observed, the phenomenon of saltatory conduction has been described [102], [194] and modeled in the literature on several occasions [146], [89], [195], [85], [95]. However, for the efficient simulation of impulse neural circuits, which are very complex circuits in the central and peripheral nervous system, reduced order models should be developed, able to accurately reproduce the saltatory conduction in low simulation times.

We propose a method (Fig. 5.2) that concatenates reduced models of myelinated compartments with 0D models of Ranvier nodes, with accuracy control. The myelinated compartments are replaced with reduced models from a hierarchical series of spatially distributed linear models developed before and described in Section 4 [66]. For the Ranvier nodes we use the famous nonlinear 0D model (Hodgkin-Huxley – HH) described below. The resulted model reproduces the saltatory conduction with acceptable accuracy. The global model, which is also nonlinear, is further subjected to a reduction procedure and the reduced model of a myelinated axon is generated.

5.1 Modeling of components

5.1.1 Modeling of myelinated compartments

The myelinated compartments are replaced with a reduced model from a hierarchical series of models developed before and presented in Chapter 4. This series contains 9 types of models, of three spatial geometry classes: 2.5D, 1D and 0D. In each class there are three categories of models: analytical, numerical and reduced order models. The series is hierarchized based on modeling errors, which are closely related to the complexity of the models [193].

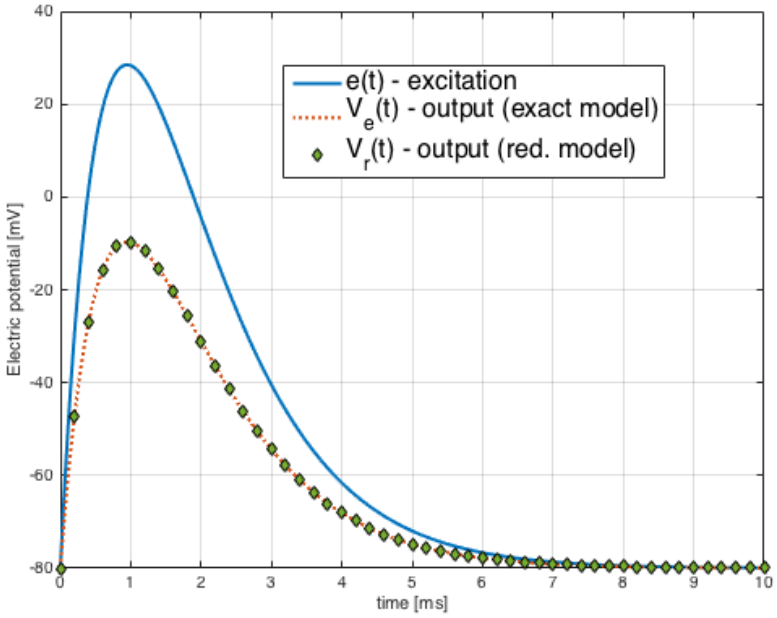


Figure 5.3: The electric potential of a myelinated compartment: excitation, at far end before the reduction, at far end after the reduction [193].

The best model from this series proved to be the analytical 1D model reduced with the vector fitting (VF) technique [164]. Different lengths were considered for the myelinated compartment: $0.25\lambda_0$, λ_0, $2.5\lambda_0$, with $\lambda_0 = 0.215$ mm representing the characteristic length (the length constant of the line). Figure 5.3 shows the electric potential at the end of the compartment of length λ_0 before and after reduction (order 3). The excitation potential $e(t)$ is approximated with an expression of two exponentials $e(t) = V_0 + V_m(e^{-t/\tau_1} - e^{-t/\tau_2})$, with $V_0 = -80$ mV, $V_m = 2800$ mV, $\tau_1 = 1$ ms, $\tau_2 = 0.9$ ms. The potential decreases in amplitude as it diffuses along the insulated compartment. The compartment model used in what follows corresponds to length λ_0 and order 3.

5.1.2 Non-linear 0D models of the Ranvier nodes

The simplified modeling of the Ranvier nodes membrane has had an intense scientific interest, so that there are several non-linear 0D models, of which the most commonly used are: FitzHugh-Nagumo (FHN) [99], Frankenhaeuser-Huxley (FH) [100], Izhikevich (Iz) [101]. These models can be regarded as low-order approximations of the highly non-linear Hodgkin-Huxley (HH) model [98], and are preferred in theoretical studies, precisely because of their relative simplicity. However, these non-dimensional reduced models are not able to retain the physical and biological significance of the inner parameters. For this reason in this case study the Ranvier nodes in this study are modeled with the HH model.

The mathematical HH model consists of four nonlinear ODEs, in which one describes a linear capacitive effect, having as state quantity the membrane voltage V, and the other three characterize the voltage-gated channels opening, by the state variables n, m and h. They are dimensionless quantities between 0 and 1 that describe the potassium channel

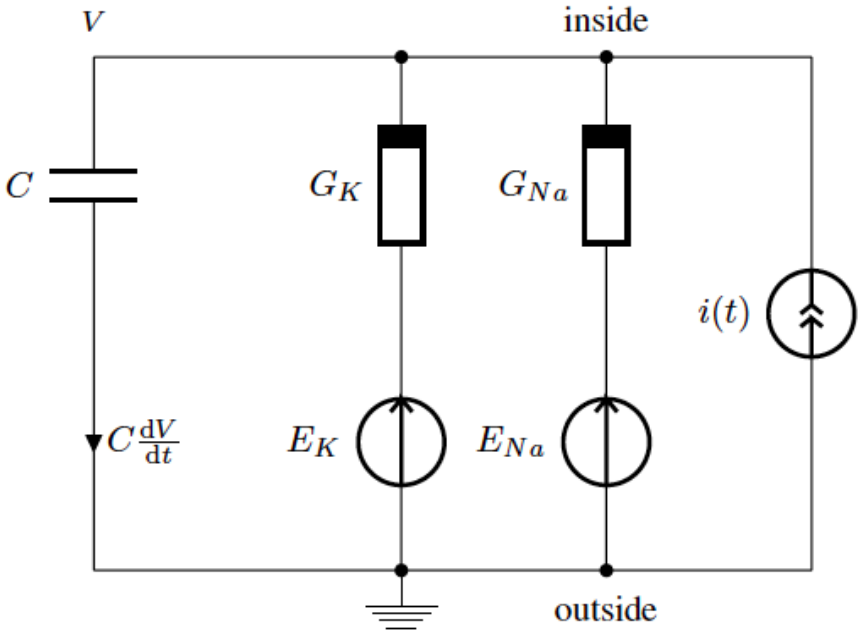


Figure 5.4: Equivalent circuit for the Hodgkin-Huxley model of a Ranvier node.

activation (n), sodium channel activation (m) and sodium channel inactivation (h).

$$\begin{aligned} C\frac{\mathrm{d}V}{\mathrm{d}t} &= -G_K(V-E_K) + -G_{Na}(V-E_{Na}) + i(t) \\ \frac{\mathrm{d}n}{\mathrm{d}t} &= \alpha_n(V)(1-n) - \beta_n(V)n \\ \frac{\mathrm{d}m}{\mathrm{d}t} &= \alpha_m(V)(1-m) - \beta_m(V)m \\ \frac{\mathrm{d}h}{\mathrm{d}t} &= \alpha_h(V)(1-h) - \beta_h(V)h \end{aligned} \tag{5.1}$$

The equivalent circuit described by (5.1) is shown in Fig. 5.4. The outside of the cell is considered to have null potential, therefore the only state variables are the potential inside the cell V and the gating variables n (activation variable for potassium channel), m (activation variable for sodium channel) and h (inactivation variable for sodium channel).

- The quantity C is a constant value representing the capacitance of the node, the corresponding capacitor being initially charged at a resting potential of a typical value of -80 mV.
- $i(t)$ is the input signal, the current that travels and reaches the node.
- G_K and G_{Na} are the conductances of the potassium and sodium channels respectively and they depend nonlinearly with respect to the node potential and the gating variables: $G_K = \bar{G}_K n^4$, $G_{Na} = \bar{G}_{Na} m^3 h$ (where $\bar{G}_K$ and $\bar{G}_{Na}$ are constants).
- The gating variables n, m and h are dimensionless quantities between 0 and 1 that describe the degree of ion channels opening. Their initial values are also known, they correspond to a resting state of the node, in which no signal is traveling: $n_0 = 0.317$, $m_0 = 0.052$, $h_0 = 0.596$.
- Coefficients α_* and β_* are rate parameters related to the steady state value of the ac-

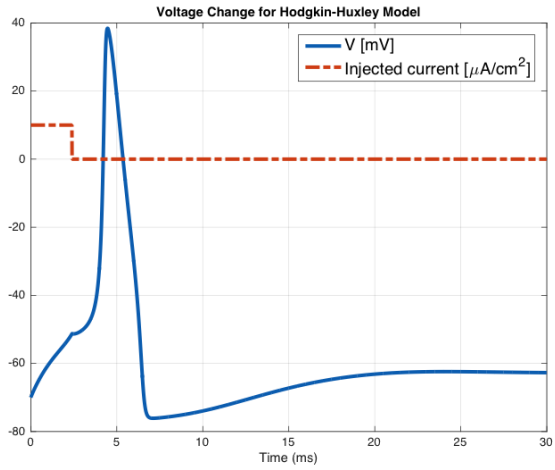


Figure 5.5: The HH model: variation of $V(t)$ and $i(t)$ = pulse of 10 μA/cm^2 for 2.4 ms ($V_0 = -70$ mV).

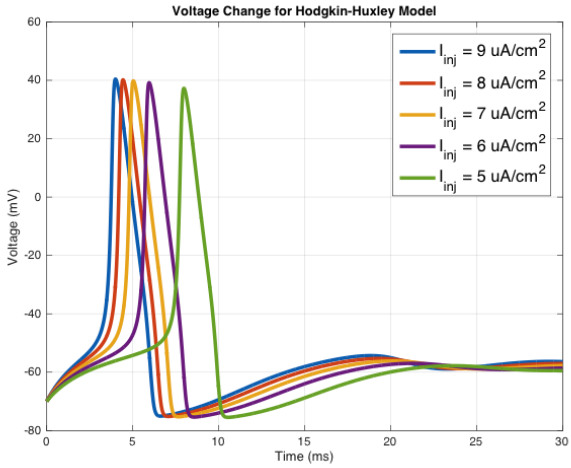


Figure 5.6: The HH model: variation of $V(t)$ for $i(t)$ step functions of different values, $V_0 = -70$ mV.

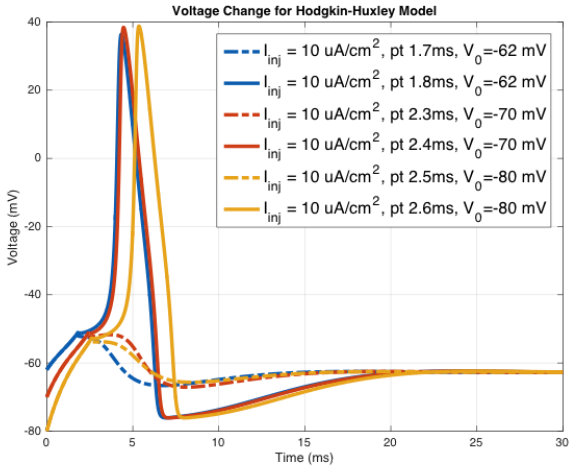


Figure 5.7: The HH model: electric potential activation for injected current $i(t)$ pulse of different durations and different initial values for V.

tivation and its time constant for reaching this value; they have known dependencies with respect to the node voltage V.

- Each ion species has a equilibrium (reversal) potential known as battery potential and denoted by E_K and E_{Na} (constant values).

The node is thus described by a compact 0D model, it involves no space variables.

The reproduction of the HH model in MATLAB and simulation of the electric potential – with numerical values for the HH parameters taken from [196] – is illustrated in the figures, as follows:

- Fig. 5.5: the variation of the electric potential V, for injected current $i(t)$ a pulse of 10 μA/cm^2 for 2.4 ms (simulation time 30 ms). The signal is delayed, as it happens in reality in the Ranvier nodes.
- Fig. 5.6: the variation of the electric potential $V(t)$, for $i(t)$ step functions of different values. The delay of the signal depends on the magnitude of the injected current (the delay is larger if the current is weaker).
- Fig. 5.7: the electric potential activation for injected current $i(t)$ pulse of different durations and different initial values for V. The action potential is not triggered if the excitation pulse does not last long enough (for a pulse of 10 μA/cm^2 and initial value $V_0 = -70$ mV, the duration has to be at least 2.4 ms).

5.2 Coupled macromodels of myelinated axons

The saltatory conduction in myelinated axons implies the simulation of a nonlinear system obtained by coupling linear models of myelinated compartments with nonlinear models of the Ranvier nodes. For an efficient simulation, model reduction is compulsory. The simulation can be done either by coupling reduced models or by formulating the problem with full models and reduce the coupled system. For both alternatives, the components can be described either as circuit netlists (and use circuit simulators, such as Spice [197]) or by using a systemic approach (which can be described in Simulink). We present here the circuit macromodel and the systemic approach.

5.2.1 Circuit coupling

The blocks in Fig. 5.8 represent sub-circuits. A chain of sections Nx-Lx are generated and the model is completed with a nonlinear bloc. Appendix A3 contains the MATLAB code for the generation of the coupled model for Spice and the resulted Spice circuit corresponding to the global mode. The circuit describing the linear blocks is extracted from the VF reduction procedure and the nonlinear circuit for the Ranvier nodes is the one in Fig. 5.4. The coupling is illustrated in Fig. 5.9. The electric potential at the output of every nonlinear node for a 6 sections (Nx-Lx) interconnection is shown in Fig. 5.10, when the left end of the axon is excited with an impulse current of 20 nA, having a width of 5 ms. The solution was obtained using the second-order implicit Modified trapezoidal

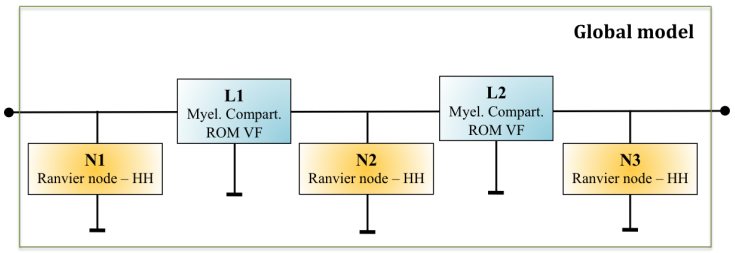


Figure 5.8: The coupling of models in the circuit macromodel. The figure represents the interconnection of 2 sections Nx–Lx. The blocks represent sub-circuits.

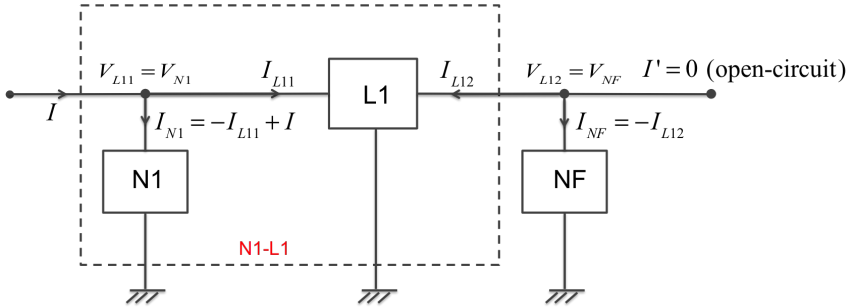


Figure 5.9: The macromodel obtained by the coupling of circuits N1–L1–NF.

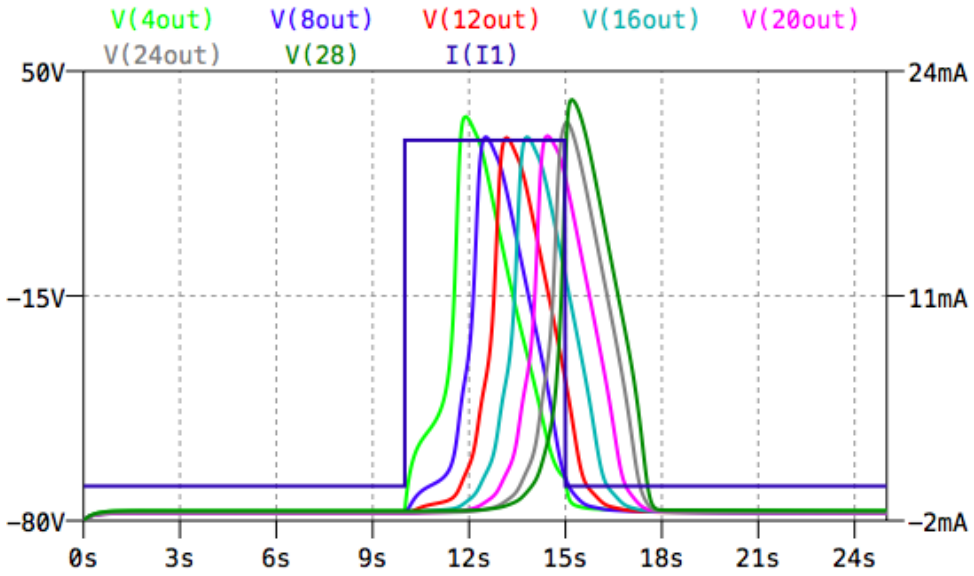


Figure 5.10: Circuit macromodel: the electric potential at the output of every nonlinear node for an axon with 6 sections Nx–Lx. The quantities are scaled: the time is in [ms], the potentials are in [mV] and the current in [nA].

integration method, with a relative tolerance of 0.001. Note that the potential is delayed at every nonlinear node. This behavior describes the phenomena of saltatory conduction and is validated with the literature [146].

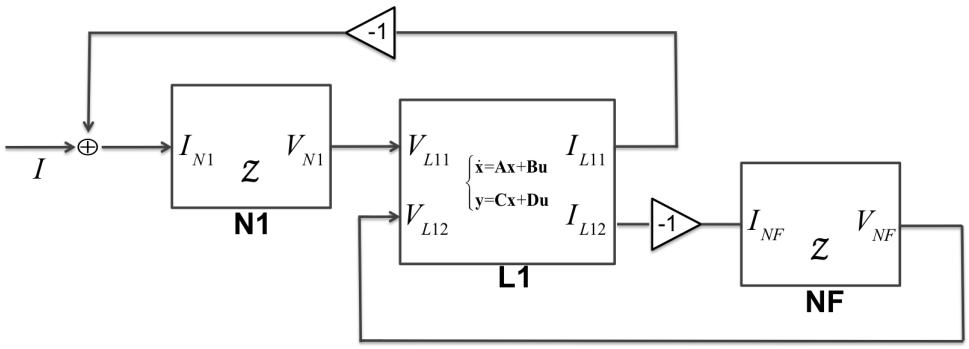


Figure 5.11: The systemic coupling for blocks N1–L1–NF.

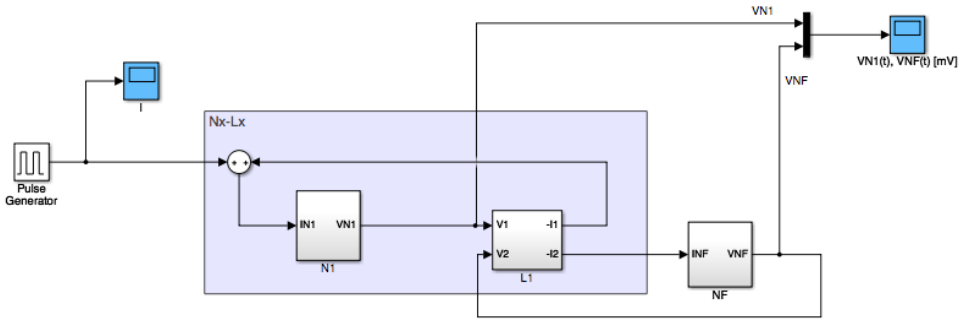


Figure 5.12: The coupled macromodel in Simulink: N1–L1–NF.

5.2.2 Systemic coupling

The nonlinear circuit can be defined as a nonlinear SISO system having as input the current and the output the electric potential V_N, described by:

$$V_N = \mathbf{Z}(I_N) \tag{5.2}$$

where $\mathbf{Z}$ is a nonlinear operator.

The linear block is a state space system, with the state space matrices ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$) extracted from the VF reduction procedure.

The coupling of a N1–L1 section (nonlinear system–linear system) with a final nonlinear block NF is illustrated in Fig. 5.11. The coupling conditions are:

- For the interconnection N1–L1:
$$\begin{cases} V_{L1,1} = V_{N1} \\ I_{N1} = -I_{L1,1} + I \end{cases}$$
- For the interconnection L1–NF:
$$\begin{cases} V_{L1,2} = V_{NF} \\ I_{NF} = -I_{L1,2} \end{cases}$$

The global system obtained has as input the current I and as output the electric potential V_{NF}, while we are also interested in the electric potential at the output of every nonlinear node V_{Nx}. This system implemented in Simulink/Matlab is shown in Fig. 5.12. The blocks are sub-systems, where Nx are modeled as nonlinear systems and the time integration is performed in a Matlab procedure and L1 is described by the state space

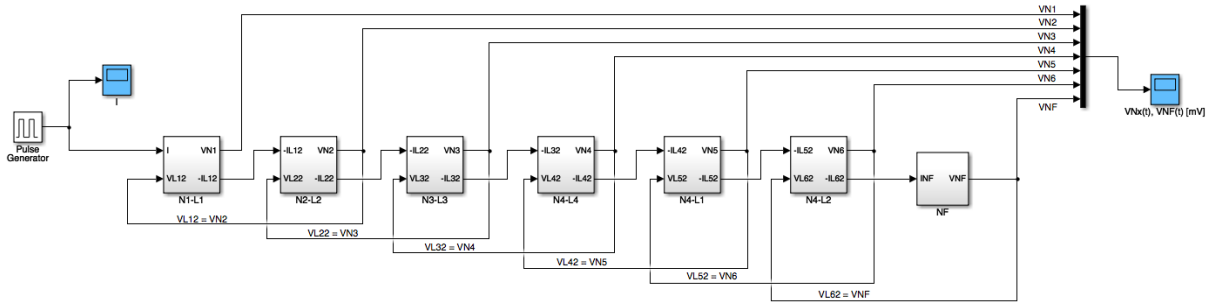


Figure 5.13: The macromodel in Simulink, with 6 coupled sections: (6× Nx–Lx)–NF.

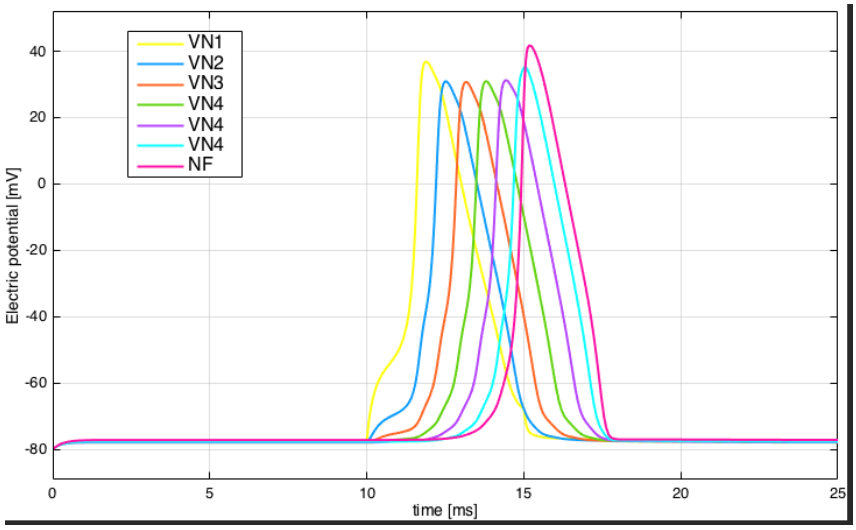


Figure 5.14: Systemic macromodel: the electric potential at the output of every nonlinear node for an axon with 6 sections Nx–Lx.

matrices of the reduced order model extracted from VF.

This coupling formulation can be generalized for a chain of many Nx–Lx sections connected with a final NF system. Then for an inner node k the coupling conditions are:

$$\begin{cases} V_{L(k-1),2} = V_{Lk,1} = V_{Nk} \\ I_{Nk} = -I_{L(k-1),2} - I_{k,1} \end{cases}$$

to which the coupling conditions at the beginning and end of the chain are added:

$$\begin{cases} V_{L1,1} = V_{N1} \\ I_{N1} = -I_{L1,1} + I \\ V_{Lx,2} = V_{NF} \\ I_{NF} = -I_{Lx,2} \end{cases}$$

The model with 6 sections Nx–Lx implemented in Simulink is shown in Fig. 5.13. The electric potentials at the output of every nonlinear block (including the last) are shown in Fig. 5.14. This behavior is validated qualitatively with the circuit model in the previous formulation. The solver used is ode15s from Matlab, with variable-step and a relative tolerance of 0.001.

The two macromodels of the saltatory conduction in myelinated axons are obtained by concatenating reduced order models of the myelinated compartments with nonlinear

models of the Ranvier nodes. The first is a circuit macromodel, implemented in a circuit simulator. In this case, the equivalent circuit for the linear blocks has to be generated by the reduction procedure. Chains with a large number of sections can be easily generated afterwards, with the help of short codes. Yet the success of the solving method for the global model is restricted by the facilities offered by the circuit simulator.

The systemic formulation is more advantageous because one could use directly the reduced order models obtained from the model order reduction procedure, without the need to realize them as circuits and thus artificially increase the degrees of freedom of the circuit model, as usually this realisation includes a lot of controlled sources. Moreover, in such an approach the user has direct access to the library of available robust nonlinear ODE solvers. However, in this approach procedures to couple the models are required and they have to be tailored for the specific significance of the input/output signals that are considered for each constitutive part. Also, the lack of code generation makes this model hard to scale up to tens or hundreds of Nx-Lx sections. In what follows we apply the reduction procedure for the circuit macromodel.

5.3 Reduction of full axon models

The model reduction uses Proper Orthogonal Decomposition (POD), a data-oriented reduction method based on processing solution samples at different time moments. This approach is suitable to reduce nonlinear systems as it only analyzes the behavior towards a particular input signal.

In our case the matrix $\mathbf{X}$ contains samples for every state variable (V, n, m, h) from the initial model at different time moments, resulting in a dimension of 56×556 and rank 56. The full model is reduced to models of orders 1 to 10; the responses are shown in Fig. 5.15 for a simulation time of 30 ms. The relative error (Fig. 5.16) is computed as follows:

$$\varepsilon_{rel} = \frac{\varepsilon_{abs}}{|\max(F_e)|} = \frac{1}{|\max(F_e)|}\sqrt{\frac{\sum_k |Fe_k - Fa_k|^2}{N}}, \tag{5.3}$$

where F_e and F_a represent the responses before (exact) and after (approximated) truncation, respectively. The error drops under 10% from order 7 (relative error of 7.88%).

The exact same results are obtained for all reduction orders for a matrix containing only the electric potential V samples, meaning a matrix $\mathbf{X}$ of size 14×556. This suggests that a proper reduction can be performed without needing information about all the state variables from the initial model, which is advantageous for longer axons, with a larger number of Ranvier nodes.

A real axon can however reach a length of 1 meter and the myelinated compartments have around 2 mm [69], [198], [146]. Neglecting the length of the nonlinear node leads to axons with up to 500 Nx-Lx sections. The reduction details for models with 100 and 500 Nx-Lx sections are given in Table 5.1. The simulation of 320 ms for the model with 500 sections took 15 minutes on a two-core 3GHz, 2GB of RAM, whereas the reduction of this global model to a model of order 70 needed under 30 seconds. The time responses are shown in Fig. 5.17. Errors smaller than 10% are obtained from order 10 for the 100

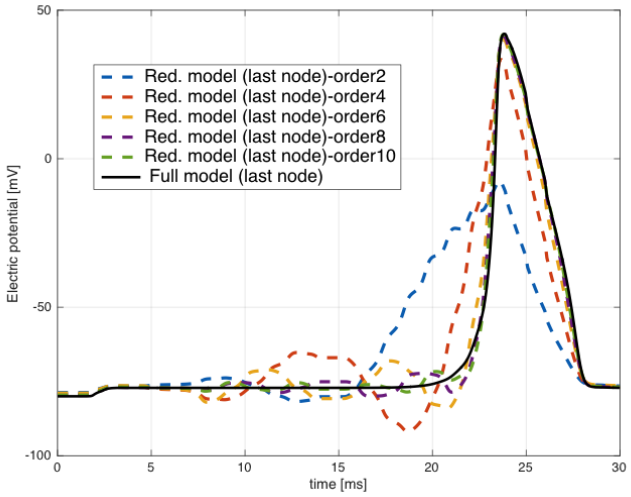


Figure 5.15: The response of the full and reduced models of orders $1 \div 10$. The displayed voltage corresponds to the last Ranvier node in the chain (the 14^{th}) [193].

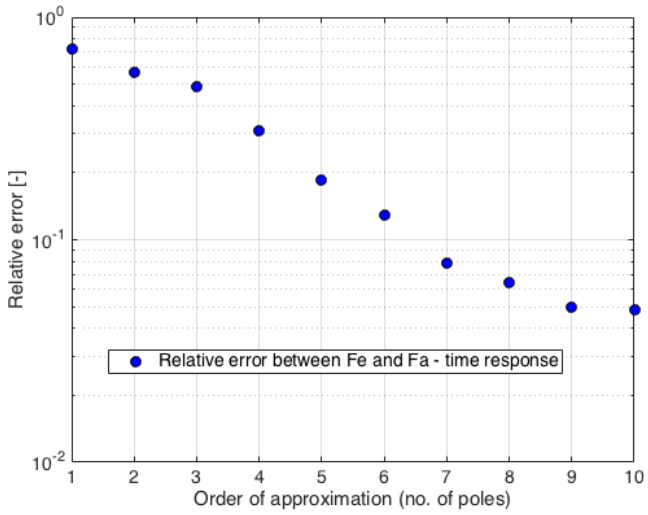


Figure 5.16: The relative error of the full and reduced models of orders $1 \div 10$ [193].

Table 5.1: Details of the model reduction with POD for global models with 100 and 500 sections.

Nx-Lx sections	Period simulated	Size of $\mathbf{X}$	Rank of $\mathbf{X}$	Order at which errors drop below 10%
13	30 ms	14×556	13	7
100	250 ms	101×1648	100	10
500	320 ms	501×2049	500	63

sections chain and from order 63 for the 500 sections chain. The results improve if the period simulated, from which the samples matrix is extracted, is larger [193] (but this means a higher offline cost) and/or we use a non-uniform snapshots grid, denser around the spike and sparser in the rest (the spike is differently located for every Ranvier node). The reduction procedure code in MATLAB is available in Appendix A3.

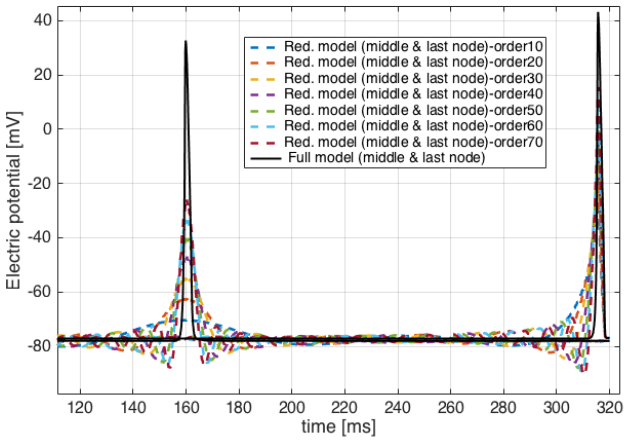


Figure 5.17: The response of the full (500 sections Nx-Lx) and reduced models of orders $1 \div 70$. The displayed voltages correspond to the middle and the last Ranvier node in the chain (the 250^{th} and the 501^{st}) [193].

This chapter has presented a method for the efficient modeling of saltatory conduction in neuronal axons. The main function of an axon is the transmission of information. The saltatory conduction is a proof for the optimality of the myelinated fiber. In order to efficiently simulate complex circuits, it is important to find the equilibrium between the complexity and the accuracy of the comprised models. The extracted model is able to reproduce the saltatory conduction with controlled accuracy. The myelinated compartments are selected from the series based on the imposed modeling error and the Ranvier nodes are modeled with HH zero-dimensional nonlinear model. The hierarchy of myelinated compartments ordered by their modeling error allows the control of accuracy, which is closely related to the models' complexity. The 0D nonlinear models are able to regenerate the signal, so that the resulting reduced model gives control on the inner model parameters (geometrical data, material constants, excitation type and value, 0D system parameters). This model is a basis for more complex simulations, of the electric potential measured in the extracellular 3D space of axons and neural circuits.

Chapter 6

Transcranial Magnetic Stimulation

This chapter presents the modeling of Transcranial Magnetic Stimulation (TMS). The human head is first modeled as a homogenous conductive sphere and a procedure is proposed, to extract a hierarchy of models of varying accuracy for TMS based on MG-EC(t) field. The second part of this chapter describes a pipeline for the generation of a 3D geometrical model from medical imaging data. The proposed approach for TMS modeling is extended for the geometrically realistic 3D model of the human head. This chapter is the development of the ideas presented in [64].

6.1 TMS, Induced Field

TMS is a noninvasive medical procedure that consists of bringing a coil near the scalp. There is a current that flows in the coil, which can be pulses of low frequency, sinusoidal or non-sinusoidal. The time variable magnetic field produced by this winding penetrates the skull where an electric field is induced. Consequently, eddy currents are generated, which overlap the natural neural signals and influence the brain functioning. A coherent activation of neurons is generated in the stimulated area, as well as in other areas by synaptic transmission. At a microscopic level, the induced electric field affects the transmembrane potential in neurons and, consequently, it influences the ion channels. This alters the metabolism and blood flow, which is visible by medical imaging tools [64].

TMS has now become an important tool for early diagnosis and treatment of neurological and psychiatric diseases [199], in the treatment of depression through deep transcranial magnetic stimulation (dTMS) [200]. In the last decade, the number of applications of the TMS procedure increased dramatically and there are new paradigms concerning its simulation, as well as technical advances that have led to new devices and integrators of TMS with EEG, PET and fMRI [201]. TMS can provide information on motor cortex excitability, on the functional integrity of neural inter-cortical structures, or corticospinal conduction. It is also used in neurosurgery for preoperative evaluation, in order to establish optimized surgical procedure. During the pre-surgical planning, the risk of post-surgical deficiencies is minimized by identifying the areas responsible for language and movement [202]. In [203] a noninvasive technique of combining TMS with EEG is

presented, which correlates the evoked potentials by TMS with the activity described by EEG in order to study the connectivity.

The correct understanding of the effects of TMS implies computer modeling to determine the induced field/currents in different anatomical structures of the brain by several coil types. Clinical experience has shown that TMS procedure has positive therapeutic effects, if applied properly with respect to duration, intensity, type of time variation of excitatory currents, shape and position of the coils. Since there is no coil configuration capable to produce both deep and shallow focused field, clinical applications search for a balance between these two conflicting requirements [204]. Several coil shapes were studied [205], [206], such as: circular coils, 8-shaped coils (field focus properties) [207], [208], H-coil type (with best ability to produce profound stimulation without affecting superficial cortical areas [209]), arbitrary coil shapes [210].

The magnetic field can be calculated by curvilinear integrals along the wire of the winding [210], wire that is approximated by a polygonal line as in [211]. The head has been modeled as a sphere in [210], with three layers of different conductivity, which represent the scalp, skull, and brain [207]. Realistic models of the skull are used in [204], [212], [213], offering the possibility of validation with experimental data and also to analyze the effect of the geometry and shape of tissues and their conductivity on the induced field [212]. Since these models are patient dependent, a balance between generality and relevance should be found. In order to adequate the general procedure to a particular patient [199] proposed the use of a accurate solver, based on the 3D impedances method (IM) based on decomposition of volume extracted from MRI in rectangular cells described by electrical circuits similar to that of finite integral method, with edge impedances that depend on local material properties placed on each edge of the orthogonal grid/cells.

Most numerical models of TMS use MG+EC(t) regimes of the electromagnetic field [200], where the secondary electric field is computed using classical numerical methods, such as Finite Element Method (FEM) [204], [208], [214], Finite Difference Method (FDM, IM) [199], or Boundary Element Method (BEM) [212], [215].

6.1.1 The geometrical model

The simplest model of the head was considered, i.e. a homogenous conductive sphere of radius 8.5 cm, with a conductivity of 0.33 S/m, as in [206], [216]. For the excitation winding we consider three test cases: 1) a circular coil (Fig. 6.1), 2) a double circular 8-shaped coil, 3) a winding with three circular sections, two forming figure eight and the third placed perpendicular with a common tangential point. Each winding has a radius of 23 mm, is placed 10 mm above the scalp and it carries a current varying with a speed of 25 A/μs as in [207]. We assume each coil rotated from the z-axis with angle β, small enough so not to collide with the skull. In general, the wire is a curve Γ described by its parametric equation:

$$\mathbf{r}_s = f(t), f : (t_{min}, t_{\max}) \to \mathbb{R} \tag{6.1}$$

The function f gives, for each value of parameter $t = \theta$, the position of the point on

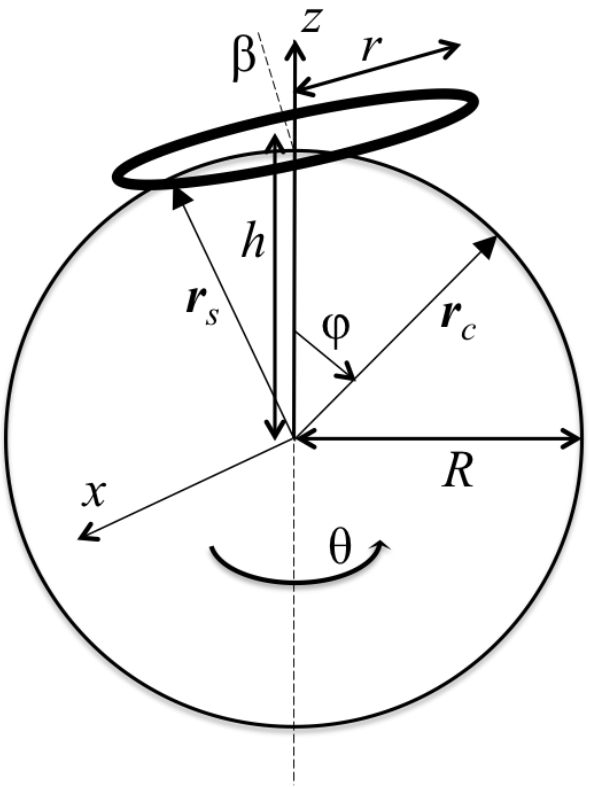


Figure 6.1: The geometry of the first test case [64].

the curve. The head's Σ is described by the function:

$$\mathbf{r}_c = F(u,v), F : (u_{\min}, u_{\max}) \times (v_{\min}, v_{\max}) \to \mathbb{R}^3 \tag{6.2}$$

Denoting by: r – the radius of the circular coil, R – the radius of the sphere, h – the height from the sphere center to which the coil is placed, the implicit equations for the two varieties are:

$$\begin{aligned} &\Gamma : x^2 + y^2 = r^2; z' = 0 \\ &\qquad \beta : x = x',\ y = y'\cos(\beta),\ z = h + y'\sin(\beta) \\ &\Sigma : x^2 + y^2 + z^2 = R^2 < r^2 + h^2 \end{aligned} \tag{6.3}$$

Using the spherical coordinates, $v = \theta$ (azimuthal angle) and $u = \varphi$ (zenithal angle), the parametric equations of the two varieties are [217]:

$$\Gamma : \begin{cases} x = r\cos(\theta) \\ y = r\sin(\theta)\cos(\beta) \\ z = h + r\sin(\theta)\sin(\beta) \end{cases} ; \theta \in (0, 2\pi) \tag{6.4}$$

$$\Sigma : \begin{cases} x = R\cos(\theta)\sin(\phi) \\ y = R\sin(\theta)\sin(\phi) \\ z = R\cos(\phi) \end{cases} ; \theta \in (0, 2\pi), \phi \in (0, \pi) \tag{6.5}$$

6.1.2 The physical model

Since the fields have a variation in time of relatively low speed, the dynamical effects can be neglected. That is why the distributions of both magnetic and electric field can be

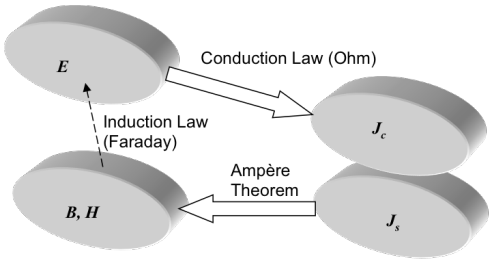


Figure 6.2: The diagram of causal relationships in MG+EC(t) approach.

computed as in a steady state regime, without eliminating the effect of electromagnetic induction law, according to which the time varying magnetic field induces an electric field.

The problem to be studied is therefore a sequence MG-EC(t), in which the magneto-steady state regime (MG) determines the distribution of the magnetic field caused by excitatory currents. The distribution of induced currents in the head is determined using equations of electro-conductive regime (EC), by neglecting their magnetic effects. This approach is simpler than that of the Magneto-Quasi-Static regime (MQS) since the induced currents do not influence the distribution of the magnetic field [64].

The causal diagram in Fig. 6.2 shows that unlike the MQS regime, in the case of MG+EC(t), the source current density $\mathbf{J}_s$ and the induced current density $\mathbf{J} = \mathbf{J}_c$ are different (one is a known source and the other is unknown). Consequently the diagram is fluent, without cycles. Thus, the solving of PDEs is simplified, since the equations to be solved are elliptic and not of parabolic type [64].

6.1.3 The mathematical model. Analytical solution.

The mathematical model is based on the first order fundamental equations in differential form of the MG field regime given by (6.6) and EC(t) regime supplemented with electromagnetic induction law (6.7). Assuming that the TMS system does not have any ferromagnetic parts for field strengthening, the space is homogeneous from the magnetic point of view, with permeability equal to that of vacuum. The EC equations are valid within the conductive domain Ω, assumed homogeneous from the conduction point of view, so with constant conductivity.

$$\begin{cases} \operatorname{div} \mathbf{B} = 0; \\ \operatorname{curl} \mathbf{H} = \mathbf{J}_s; \\ \mathbf{B} = \mu_0 \mathbf{H}. \end{cases} \tag{6.6}$$

$$\begin{cases} \operatorname{div} \mathbf{J} = -\frac{\partial \rho}{\partial t}; \\ \operatorname{curl} \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}; \\ \mathbf{J} = \sigma \mathbf{E}. \end{cases} \tag{6.7}$$

Inside the EC domain the charge is relaxed, so its volume density is null, which makes the current density solenoidal ($\operatorname{div} \mathbf{J} = 0$). Using the magnetic vector potential $\mathbf{A}$, with

Coulomb gauge, a Poisson's equation is obtained for $\mathbf{A}$:

$$\begin{cases} \mathbf{B} = \operatorname{curl} \mathbf{A}; \\ \operatorname{curl} \mathbf{H} = \mathbf{J}_s \Rightarrow \operatorname{curl} \operatorname{curl} \mathbf{A} = \mu_0 \mathbf{J}_s \quad ; \operatorname{div} \mathbf{A} = 0 \Rightarrow \Delta \mathbf{A} = -\mu_0 \mathbf{J}_s; \\ \mathbf{H} = \mathbf{B}/\mu_0 = \operatorname{curl} \mathbf{A}/\mu_0. \end{cases} \tag{6.8}$$

Then the induced field is:

$$\begin{gathered} \operatorname{curl} \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \Rightarrow \operatorname{curl} \left(\mathbf{E} + \frac{\partial \mathbf{A}}{\partial t} \right) = 0; \\ \mathbf{J} = \sigma \mathbf{E} \Rightarrow \mathbf{J} = -\sigma \left(\operatorname{grad} V + \frac{\partial \mathbf{A}}{\partial t} \right), \text{ with } \mathbf{E}' = -\frac{\partial \mathbf{A}}{\partial t} \text{ and } \mathbf{E}'' = -\operatorname{grad} V \\ \operatorname{div} \mathbf{J} = 0 \Rightarrow \operatorname{div} \left(\sigma \operatorname{grad} V + \sigma \frac{\partial \mathbf{A}}{\partial t} \right) = 0 \Rightarrow \Delta V = 0. \end{gathered} \tag{6.9}$$

where $\mathbf{E}'$ is the primary component of the electric field (solenoidal) and $\mathbf{E}''$ is the secondary component (Coulombian, non-rotational). In homogeneous domains, the secondary field is described by a harmonic scalar potential V, the solution of Laplace equation.

Problem formulation

The field problem is reduced to solving the Laplace equation inside the conductor, with Neumann boundary conditions:

$$\begin{gathered} \operatorname{div}_s \mathbf{J} = -\frac{\partial \rho_s}{\partial t} \Rightarrow \mathbf{n}_{12} \cdot (\mathbf{J}_2 - \mathbf{J}_1) = -\frac{\partial \rho_s}{\partial t} \Rightarrow \mathbf{n} \cdot \mathbf{J} = -\frac{\partial \rho_s}{\partial t} \\ \Longrightarrow \sigma \mathbf{n} \cdot \left(\frac{\partial \mathbf{A}}{\partial t} + \operatorname{grad} V \right) = -\frac{\partial \rho_s}{\partial t} \Rightarrow \sigma \frac{\mathrm{d}V}{\mathrm{d}t} = -\frac{\partial \rho_s}{\partial t} - \sigma \frac{\partial (\mathbf{n} \cdot \mathbf{A})}{\partial t}. \end{gathered} \tag{6.10}$$

Assuming there are no external charged bodies, the human body charge is null, the superficial charge density is negligible, and the boundary condition of Neumann type becomes as in [212]:

$$\frac{\mathrm{d}V}{\mathrm{d}n} = -\frac{\partial (\mathbf{n} \cdot \mathbf{A})}{\partial t} = -\frac{\partial A_n}{\partial t} = E'_n. \tag{6.11}$$

In order to have a unique solution, a reference point on the boundary is chosen. If the problem is homogeneous and axisymmetric (i.e. $\beta = 0$) then the magnetic vector potential is oriented tangentially to the surface, so its normal component is zero and therefore the solution of Laplace equation is null so there is no secondary electric field ($\mathbf{E}'' = 0$).

The primary component of the induced field: magnetic vector potential A

The solution of equation (6.8) valid for the entire 3D space is given by the Biot-Savart-Laplace integral on the domain D, in which the source current flows. If the coil is a wire, the domain D reduces to a curve Γ and:

$$\mathbf{A}(\mathbf{r}_c, t) = \frac{\mu_0}{4\pi} \int_D \frac{\mathbf{J}_s(\mathbf{r}_s, t)}{|\mathbf{r}_s - \mathbf{r}_c|} \mathrm{d}v_s = \frac{\mu_0 i(t)}{4\pi} \int_\Gamma \frac{\mathrm{d}\mathbf{r}_s}{|\mathbf{r}_s - \mathbf{r}_c|}. \tag{6.12}$$

The time derivative of this vector potential is, according to (6.9), the primary induced field.

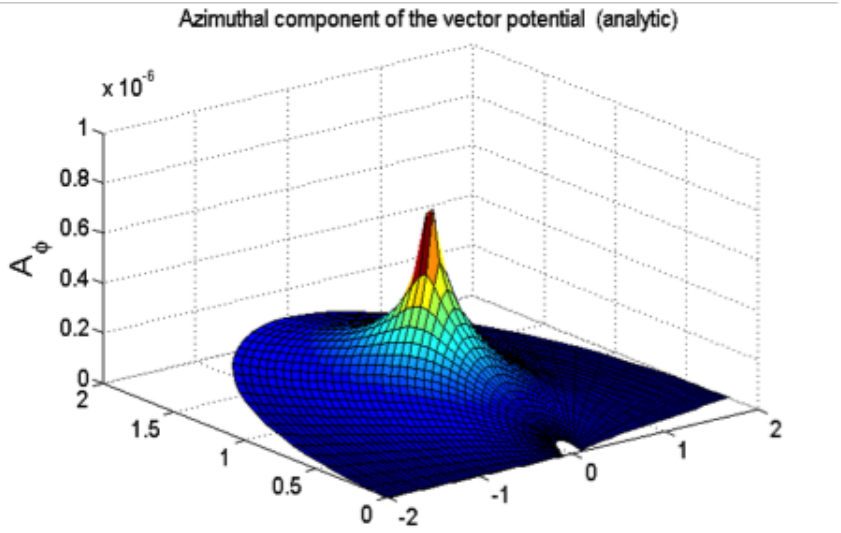


Figure 6.3: The magnetic potential generated by a circular wire. A similar image is obtained by using the numerical method.

The secondary component of the induced field: the scalar potential V

The general solution of the Laplace's equation (6.9) inside a sphere can be obtained by separation of variables and it has the form [218]:

$$V(\rho,\theta,\varphi) = \sum_{l=0}^{\infty} r^l \sum_{m=-l}^{l} a_{lm} Y_l^m(\theta,\varphi). \tag{6.13}$$

The spherical complex harmonics are defined from the associated Legendre polynomials Y and are orthogonal on the surface of the unit sphere. This remarkable property allows the Fourier expansion in a series of any arbitrary L^2 function:

$$f(\theta,\varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_{lm} Y_l^m(\theta,\varphi). \tag{6.14}$$

The Fourier coefficients are:

$$\begin{aligned} f_l^m &= \int_\Omega f(\theta,\varphi) Y_l^{m*}(\theta,\varphi) \mathrm{d}\Omega \\ &= \int_0^{2\pi} \mathrm{d}\theta \left(\int_0^{\pi} \mathrm{d}\varphi \sin(\varphi) f(\theta,\varphi) Y_l^{m*}(\theta,\varphi) \right). \end{aligned} \tag{6.15}$$

Therefore, the potential inside a conductive sphere has the analytical expression given by (6.13), where the coefficients are obtained by imposing the boundary condition:

$$\begin{aligned} \frac{\partial V}{\partial n} = -\frac{\partial A_n}{\partial t} \Rightarrow \left.\frac{\partial V}{\partial \rho}\right|_{\Sigma:\rho=R} &= \sum_{l=0}^{\infty} l R^{l-1} \sum_{m=-l}^{l} a_{lm} Y_l^m(u,v) \\ \Longrightarrow a_{lm} = -\frac{1}{l R^{l-1}} \int_0^{2\pi} \left(\int_0^{\pi} \frac{\partial A_n}{\partial t} \sin(u) Y_l^{m*}(u,v) \mathrm{d}u \right) \mathrm{d}v. \end{aligned} \tag{6.16}$$

In (6.16), $A_n(u,v)$ is the normal component of the magnetic vector potential computed with (6.12).

Table 6.1: The relative error of numerical integration for different number of nodes: Err1 using quadrature, Err2 using approximation by a polygon.

n_t	10	14	18	22	26	30	34	38
Err1	$5.3 \cdot 10^{-3}$	$4.6 \cdot 10^{-4}$	$4.2 \cdot 10^{-5}$	$3.9 \cdot 10^{-6}$	$3.7 \cdot 10^{-7}$	$3.6 \cdot 10^{-8}$	$3.5 \cdot 10^{-9}$	$3.4 \cdot 10^{-10}$
Err2	$2.1 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$6.8 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$

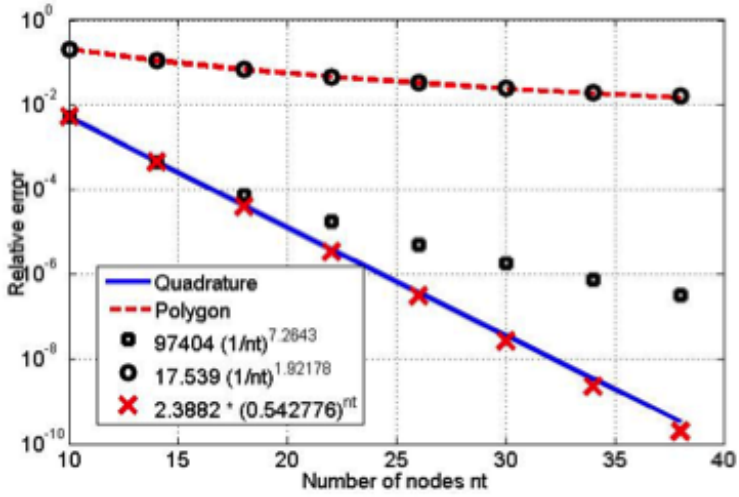


Figure 6.4: Error of the numerical integral vs. number of nodes.

6.1.4 Numerical results. Model verification.

The parameters of Γ and Σ are discretized as:

$$T = [t_1 = t_{\min}, t_2, \ldots, t_{n_t} = t_{\max}];$$
$$U = [u_1 = u_{\min}, u_2, \ldots, u_{n_u} = u_{\max}];$$
$$V = [v_1 = v_{\min}, v_2, \ldots, v_{n_v} = v_{\max}].$$

their size n_t, n_u, n_v giving the refining level of the model.

The magnetic vector potential A. The error control.

The integral (6.12) is approximated by numerical quadrature:

$$A_n(u_i, v_j, t) \cong \frac{\mu_0 i(t)}{4\pi} \sum_{k=1}^{n_t - 1} c_k \frac{\mathbf{n}_{ij} \cdot \Delta \mathbf{r}_k}{|\mathbf{r}_k - \mathbf{r}_{ij}|}. \tag{6.17}$$

The vector potential on the wire is unbounded. The numerical computation was carried out using the quadrature coefficients $c_k = 1$, as in the composite trapezoidal rule for periodic functions. The numerical values of the vector potential were checked against the analytical values, for a circular wire with uniform distributed nodes (Fig. 6.3).

Another approach to compute **A** is to approximate the wire with a polygon. The field produced by a circular wire approximated by a polygon was computed in the point on the sphere surface where the maximum field is induced. Table 6.1 and Fig. 6.4 show the convergence results of the numerical integration.

The errors were computed with respect to the analytical values of the vector potential produced by a circular coil [219]. When using quadrature, by doubling the number of

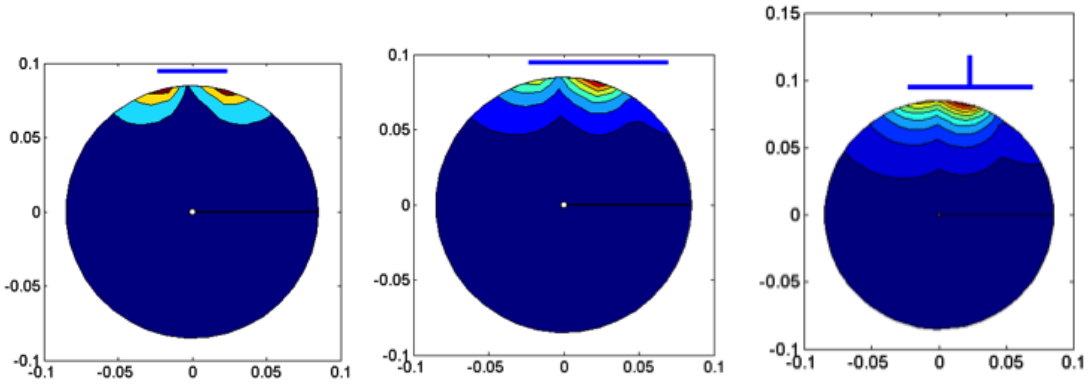


Figure 6.5: Induced field intensity in three test cases (isovalues starting at 0 – the darkest blue, with a step of 1 V/m) [64].

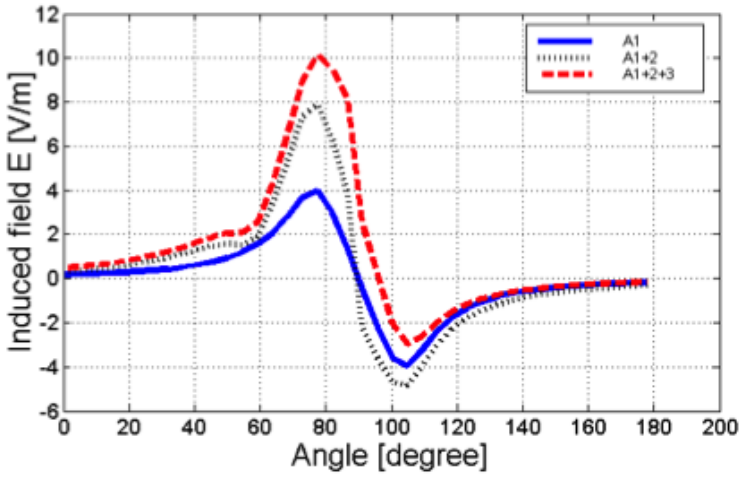


Figure 6.6: The tangential component of the primary induced electric field on the surface of the sphere, along a meridian line [64].

nodes, the relative error decreases 414 times.

This rate of convergence is impressive, with an order greater than 7. In fact the convergence is better than this with an exponential dependence $\varepsilon_{n_t} < c_1 \cdot c_2^{n_t}$, where $c_2 < 0.6$ (Fig. 6.4). This is due to the high smoothness of the function, which is infinitely differentiable. Therefore, the numerical integration achieved even with a small number of nodes, of approximately 20, gives an admissible numerical error [64].

A minimal error is expected for nodes that are uniformly distributed along the circle, because of its constant curvature. In contrast, when a curve of arbitrary shape is considered, adaptive methods will give better results. In the case of the approximation with a polygon, the error falls only 4 times when the number of nodes on a circle is doubled. Convergence seems to be of second order, as shown in the graph in Fig. 6.4.

In conclusion, the results obtained using numerical quadrature based on the composite trapezoidal rule have to be preferred to polygonal approximations proposed in [211], since that has a much better order of convergence [64]. From Table 6.1 and Fig. 6.4 it follows that the relative error is bounded by:

$$\varepsilon_{nt} < (c_1/n_t)^{p_1}, \text{ with } c_1 = 4.2, p_1 = 8. \tag{6.18}$$

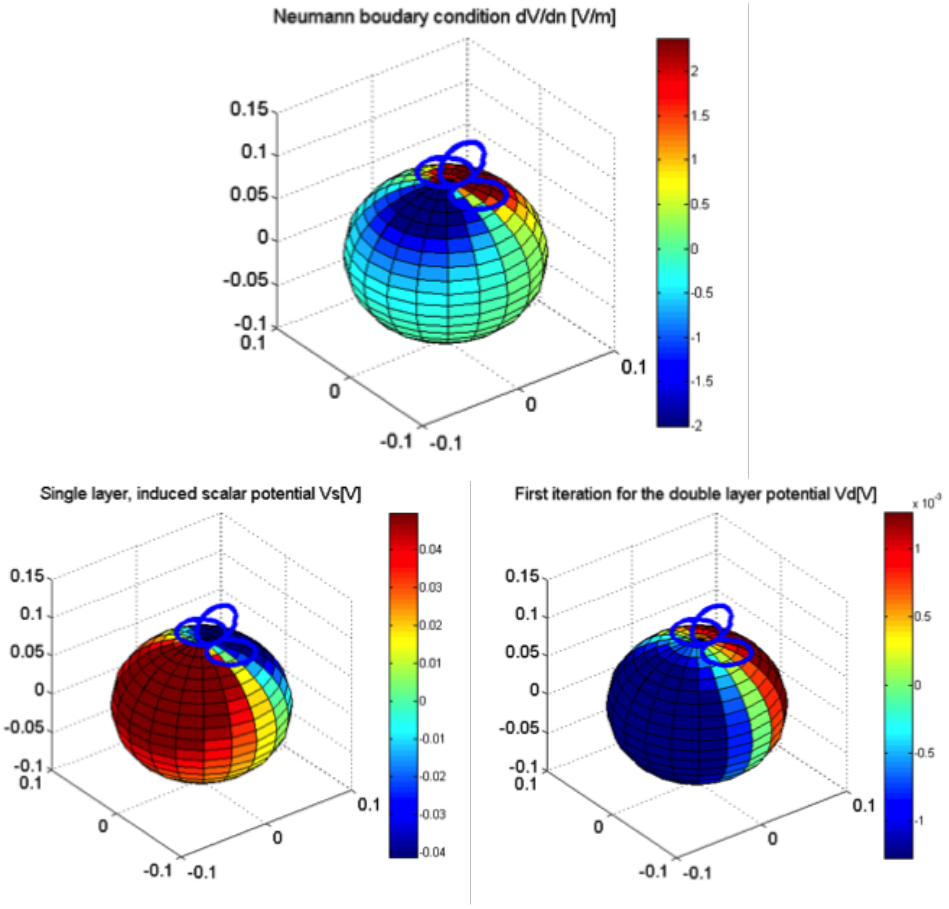


Figure 6.7: a) Variation of the normal derivative of induced scalar potential [V/m]; b)+c): Variation of potential on the surface of the sphere [64].

The induced electric field.

The strength of the primary induced electric field in the three cases studied is plotted in Fig. 6.5. Let there be noted that the third configuration induces a deeper field, while the second configuration better focuses the induced electric field.

Fig. 6.6 shows the variation of the tangential component of the electric field induced on the surface of the sphere, along a meridian line. The field has a maximum value at approximately 75 degrees from the horizontal, therefore relatively close to the point of tangency, located near the zenith. As expected, the maximum field is double in the second test case and 2.5 times higher in the third test case.

The determination of secondary induced field starts from the value $\mathrm{d}V/\mathrm{d}n$ of the Neumann boundary conditions on the sphere surface, which is proportional to the time derivative of A_n as in (6.11). Its distribution is plotted in Fig. 6.7a.

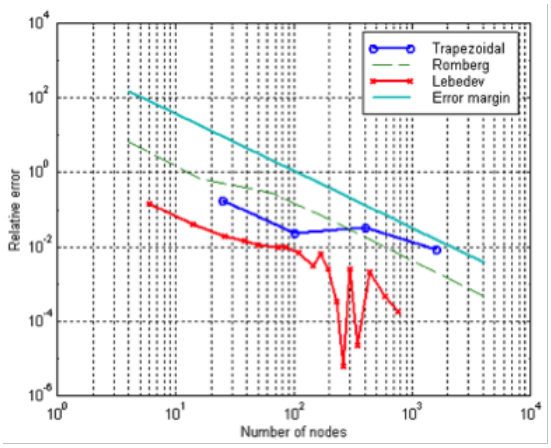


Figure 6.8: The relative error of numerical integral over the sphere [64].

The potential of the sphere's surface satisfies the BEM equation [64]:

$$
\begin{aligned}
2\pi V(P) &= \oint_{\Sigma} \left[\frac{1}{R'} \cdot \left. \frac{\partial V}{\partial n} \right|_{Q} - V|_{Q} \cdot \frac{\partial}{\partial n} \left(\frac{1}{R'} \right) \right] \mathrm{d}S' \\
\Longrightarrow 2\pi V_{il} &+ R^2 \sum_{\substack{j=1 \\ R>\varepsilon}}^{n_u-1} \sum_{k=1}^{n_v-1} \frac{\sin\theta_j \Delta\phi_k \Delta\theta_j}{R^3_{kj-il}} \left(\mathbf{R}_{kj-il} \cdot \mathbf{n}_{kj} \right) V_{kj} \\
&= -R^2 \sum_{\substack{j=1 \\ R>\varepsilon}}^{n_u-1} \sum_{k=1}^{n_v-1} \frac{\sin\theta_j \Delta\phi_k \Delta\theta_j}{R_{kj}} \cdot \left. \frac{\partial A_n}{\partial t} \right|_{kj} - \varepsilon \left. \frac{\partial A_n}{\partial t} \right|_{il} .
\end{aligned}
\tag{6.19}
$$

Fig. 6.7b shows the primary component (single layer) of the potential on the surface with the expression obtained from the integral equation of the potential [220]. If both terms of the integral equation of the potential are discretized a $(n_u \times n_v)$ system of linear algebraic equations satisfied by the potential in the nodes on the surface is obtained.

The single layer potential (Fig. 6.7b) and the double layer potential (Fig. 6.7c) have been determined for several values of n_u and n_v and the results were extrapolated with a Richardson procedure. Using this extrapolation the bound of the relative error of the numerical integration V_{maxR} was deducted:

$$
\varepsilon < (c_2/n_u)^{p_2} + (c_2/n_v)^{p_2} , \text{ with } c_2 = 15, p_2 = 2. \tag{6.20}
$$

In addition to the composite trapezoidal rule (6.19), the numerical integration was done also with Romberg [2] and Lebedev [221], [222] methods. The convergence of these approaches is shown in Fig. 6.8. The numerical integration over the sphere's surface is necessary in both analytical (6.16) and numerical – BEM (6.19) approaches. Still there is a difference, in (6.16) the function is smooth, whereas (6.19) is an improper integral. The best approach to compute it is FMM, based on exact computing [220] of close interactions and multi-pole approximation of far-away ones, e.g. $1/R$ as in [219].

6.2 TMS and realistic geometries

6.2.1 Pipeline for the generation of a 3D geometrical model from medical imaging data

Medical imaging techniques such as MRI (Magnetic Resonance Imaging) or CT (Computed Tomography) consist of non-invasive procedures with the purpose of understanding the anatomy and physiological processes of the body. MRI scanners use powerful magnetic fields, radio waves and field gradients to form images of an organ. These images represent the characteristics of the fluids in shades of gray, as follows: fats, melanin, high-protein-rich fluids are represented by open shades, whereas bones, tumors, sore areas in dark shades; in the brain, gray matter is represented with a darker shade than the white matter. More details about MR can be found at [223]. Different kinds of MRI exist, depending on their purpose: structural MRI, diffusion MRI (dMRI) or functional MRI (fMRI).

In [224] it is described a procedure for extracting a model for COMSOL, with WM (white matter), GM (gray matter) and CSF (cerebrospinal fluid) sections, which uses source files from BrainWeb (*.mnc*) database [225], [226] and the BrainSuite segmentation program [227] (BrainSuite is also used in [228]). However the procedure for extracting the 3D model is not described here.

Reference [228] starts from MR images created with a 3T scanner (Magnum 3.0, Medinus Inc., Republic of Korea). The sections segmented are scalp, skull, CSF, GM, WM. From the diffusion-weighted MR scans the tensor matrix (DT) and the fractional anisotropy map (FA) are determined. The programs used are SPM5 [229], BrainSuite (for segmentation) and CGAL [230] (mesh generation library). [214] takes into account different conductivities for the scalp, skull, brain and ventricular system.

The effect of heterogeneity and anisotropy on the distribution of field and currents induced in the brain by magnetic stimulation is also studied in [231]. The model incorporates data regarding the conductivity tensor extracted from diffusion measurements, but the head is modeled with a sphere, not with realistic 3D models.

The source files in [232] consisted of MR images of a healthy 38-year-old male subject generated with the Siemens Magneton Vision 1.5 T scanner and saved in the Analyze format. A soft simulation package was used to extract the initial geometry of the CAD model. The sections generated were skin, skull, CSF, GM, WM.

An ample study that highlights the extent to which realistic geometries are relevant for the accuracy of TMS simulation is presented in [212]. It is stated that the level of anatomic detail of a study must be large enough to highlight the secondary component of the induced electric field. For the realistic geometries it was found that the secondary component of the induced field has a weight of about 20-35% from the total induced field. In most cases the direction of the secondary field is opposite to the primary one, decreasing its value. MR (for soft tissue) and CT (for bone) images of a neurologically healthy male subject were used. The MR images (T1-weighted) were created with a 1.9 T GE/Elscint Prestige MRI (Haifa, Israel) scanner at the RIC (Research Imaging Institute)

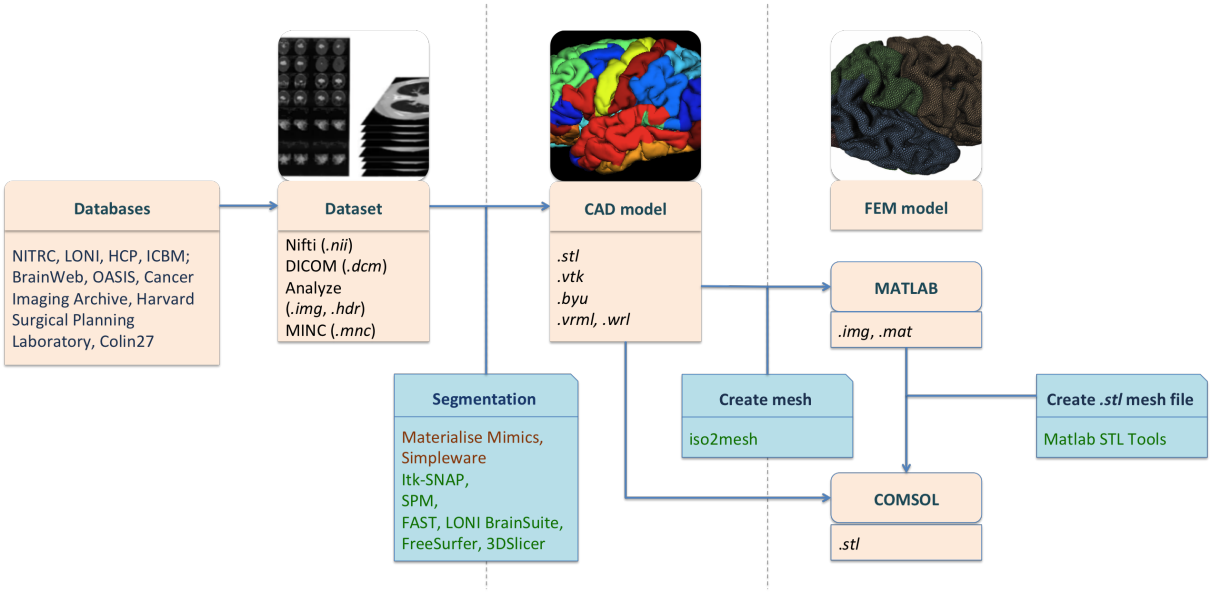


Figure 6.9: The process of creating a 3D FEM model from MR/CT files.

and the CTs were taken from the Cancer Therapy and Research Center, UTHSCSA, being previously generated with a PQ5000 CT Scanner. The segmentation was performed with FAST (FMRIb's Automated Segmentation Tool, part of the FMRIb Library). The in-house MANGO program was used to visualize the 3D images. CT images were used for the skull, segmented based on a bone threshold (greater than 1500HU). The individualized sections were scalp, skull, CSF, GM, WM and ventricles (having the same conductivity as CSF). Different conductivity values were considered for each class of tissue as follows: scalp 0.33 S/m, skull 0.0042 S/m, CSF 1.79 S/m, GM 0.33 S/m, WM 0.14 S/m, ventricles 1.79 S/m.

In [215] the MR images were obtained from a male subject using a 3T scanner (Tim Trio, Siemens Medical Solutions, Erlangen, Germany). The segmented compartments were: scalp, skull and intracranial space. The cortical surface was extracted from a MPRAGE T1-weighted volume with the FreeSurfer program [233].

The anatomical model in [204] is a 26-year-old female subject, "Ella" from the Virtual Family [234]. The MR images were segmented into 76 different tissues throughout the body. In the brain, the structures identified were the cortex, WM, cerebellum and deep brain. The dielectric properties of each tissue were selected from the literature [128], [235].

A male volunteer was used by [236], the resolutions for CT and MR being 0.488/1.0 mm and 0.5/4.0 mm respectively. For the segmentation the procedure from [233] and [237] was used. The brain was segmented into cerebellum, GM, WM, ventricular system, midbrain, brainstem, and CSF all around. For the muscles of the neck, nose, etc. the Atlas of Human Anatomy [238] was used.

Reference [239] is a review of several real anatomical models manufactured with the purpose of analyzing the effects of an impact in the event of an accident. Material data is presented for different types of tissues.

The dataset in [240] consists of MR (T1 weighted) images of a 55-year-old female subject generated at University Hospital, Jena, Germany, obtained with a 3-tesla Siemens

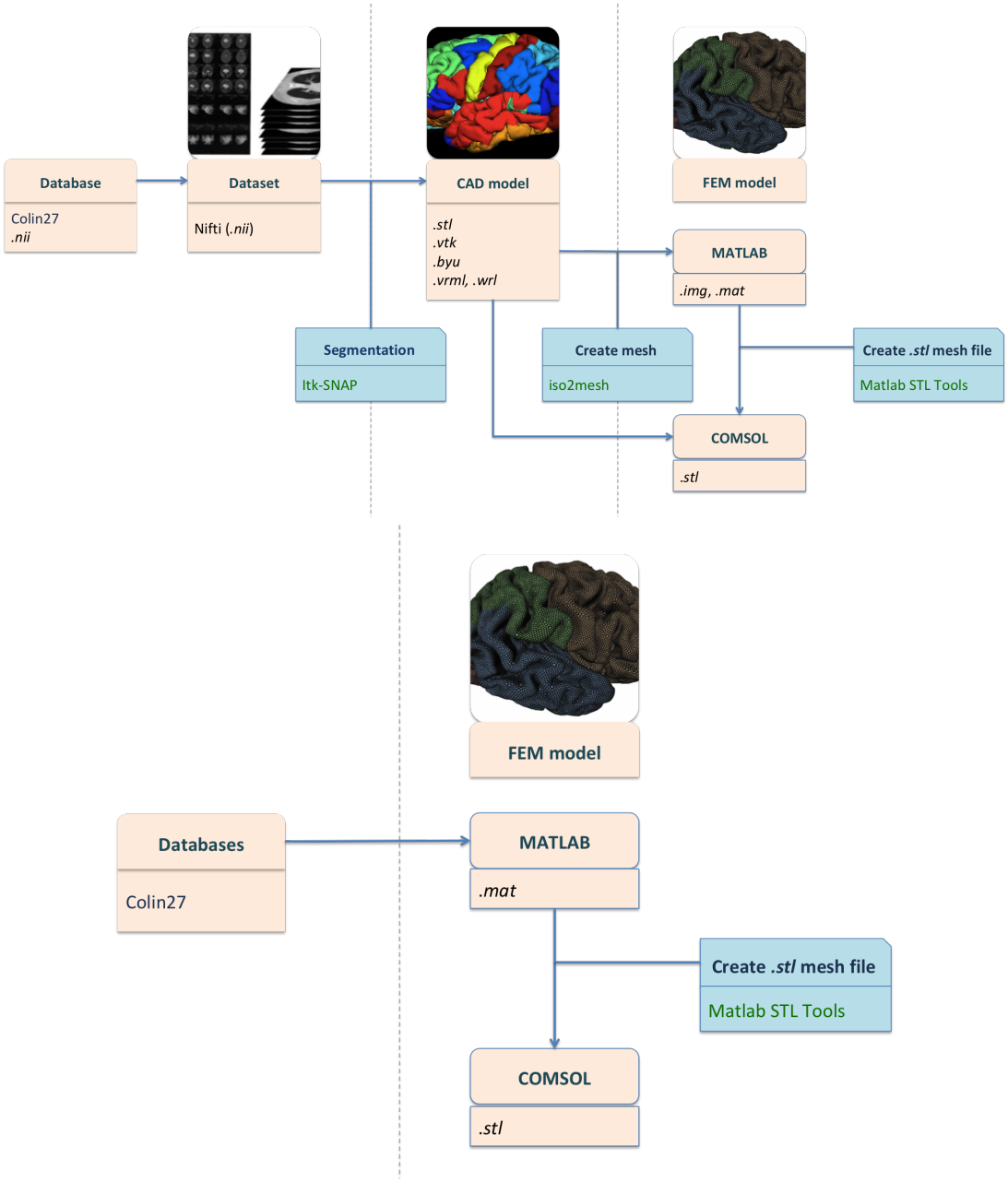


Figure 6.10: Proposed pipelines. a) using segmentation; b) using a segmented model.

MR scanner. A detailed and hybrid (automatic and manual) 3D segmentation was performed and the following tissues were identified: WM, GM, cerebellum, CSF, cortical bone, trabecular bone, dura mater, skin, eye, crystalline.

Four FEM models were extracted in [241] from MR images of the same human subject with different degrees of segmentation (with 11, 10, 9, 5 sections) for which resistivity and conductivity are given. The MR images are obtained with a 1.5 Tesla GE Signa scanner. The segmentation was performed with a semi-automated in-house program [242], then checked by a radiologist and modified accordingly.

There are dozens of programs that can be used to process MR/CT files, which encompasses 3D visualization, segmentation (partitioning and labeling), defining the regions of interest. After the creation of the so-called CAD model a 3D FEM model can be generated in a specific program such as COMSOL or MATLAB. Fig. 6.9 summarizes the steps

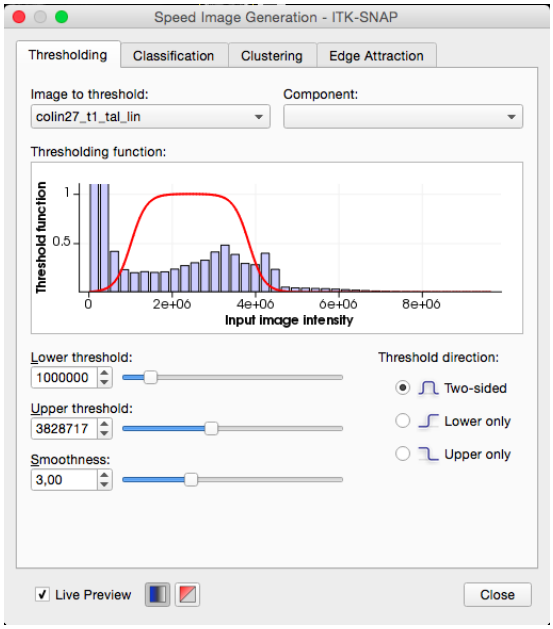


Figure 6.11: Itk-SNAP – preprocessing with thresholding.

Table 6.2: Values of thresholding indicators in itk-SNAP – pre-processing.

	Threshold	Smoothness	Smoothing curvature force
Skull	Lower = 2,300,000	2	0.1
CSF	Upper = 2,800,000	3	0.15, after 230 iterations 0.23, until 430 iterations
GM	Two-sided = 1,004,926 to 3,818,718	3	0.05 for 700 iterations
WM	Lower = 3,818,718	3	0.1 for 700 iterations

identified for generating a FEM model from MRI/CT images, the most frequently used programs, and the file types at each stage.

More information on the extraction of CAD models and on mesh generation and processing algorithms for the whole human body can be found in [243]. Details about segmentation methods and their validation are given in [244].

The visualization and analysis programs usually describe the volumes with three 2D images, representing the three planes (frontal, transversal and sagittal) in the orthogonal volume slicing [245].

The purpose of segmentation is to divide the image into a set of disjoint regions containing voxels with the same characteristics (intensity, depth, color, texture). The segmentation's result may be an image with labels for each homogeneous region or a set of contours that represent the regions' borders. In the brain, the MR images are usually segmented into three major types of tissue: white matter (WM), gray matter (GM) and cerebrospinal fluid (CSF).

Before the segmentation stage an image preprocessing is usually performed, which acts as a filter of the original image; the most used preprocessing methods (pre-segmentation) are Bias Field Correction and Image Registration – the removal of the non-brain tissue.

The segmentation methods can be divided into [246]:

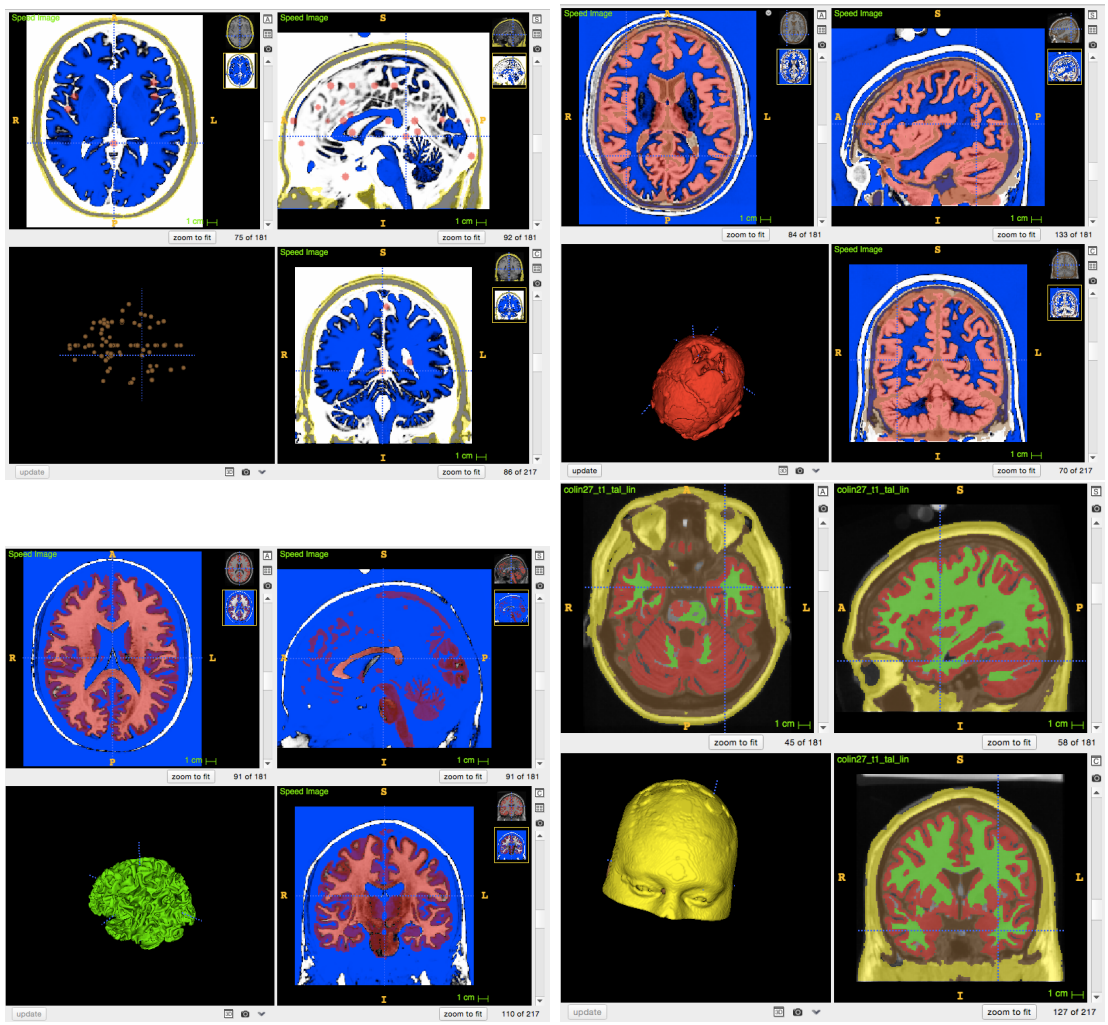


Figure 6.12: Segmentation in itk-SNAP; a) CSF, placing the seeds for region growing; b) GM after 400 iterations; c) WM after 700 iterations (final); d) all the domains.

- Manual segmentation;
- Intensity based methods (using intensity histograms, including threshold values, region growing, classification – needs Bias Field Correction, clustering);
- Atlas based methods (need Image registration), which fail if the anatomy of the subject differs significantly from the average, or if it presents deformations or abnormalities;
- Surface based methods (including active contours);
- Hybrid methods.

The two pipelines we propose for the generation of a CAD model are described in Fig. 6.10.

Pipeline 1 – Fig. 6.10a

The large number of methods and programs makes it very difficult to choose a single segmentation program. A comparison between SPM, FAST and BrainSuite is presented in [247].

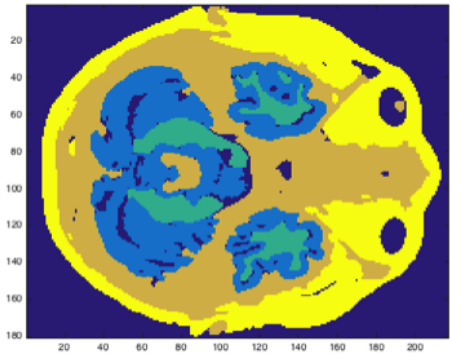


Figure 6.13: Pipeline 1 – Inner layer extracted from the segmented model; yellow: skull, brown: CSF, blue: GM, green: WM, indian blue: empty spaces.

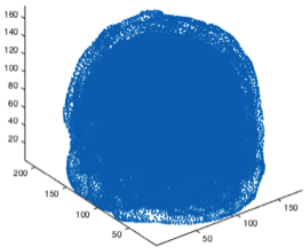


Figure 6.14: Pipeline 1 – The nodes (vertices) in the 3D mesh after segmentation.

The program itk-SNAP [245] is often used in studies, has a relatively good documentation, it is open-source and it seems intuitive for understanding MR segmentation. Unfortunately, one of the drawbacks of graphical user interface programs is that the user has no control over some (pre-)segmentation parameters. Thus, the segmentation quality depends on the available parameters and the user's ability to use them efficiently.

The segmentation in itk-SNAP is performed by associating a label to each volume element (voxel from the MR image). The union of all the images (slices) reproduces the 3D object consisting of volume elements with the same label.

STL (STereoLithography) is a 3D file format created by the company 3D Systems. The representation can be ASCII or binary. A STL file describes an unstructured surface with triangles, for each triangle having the normal and the Cartesian (3D) coordinates of the three points in the corners. STL coordinates are usually positive numbers, and the units of measurement are relative. The ASCII files can become very large (my GM file exported from itk-SNAP has almost 200 MB). COMSOL can import both STL and ASCII as well as binary files.

Preprocessing

We used a preprocessing method that offers the user more control than others: thresholding. This gave better results than the alternative method of classification provided by itk-SNAP. Thresholding involves image filtering based on the pixel intensity histogram,

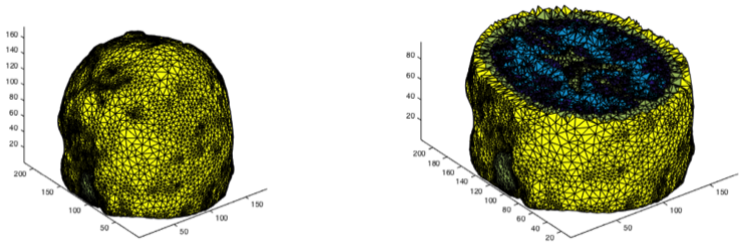


Figure 6.15: Pipeline 1 – The 3D mesh with the four segmented regions colored differently – overview and section view.

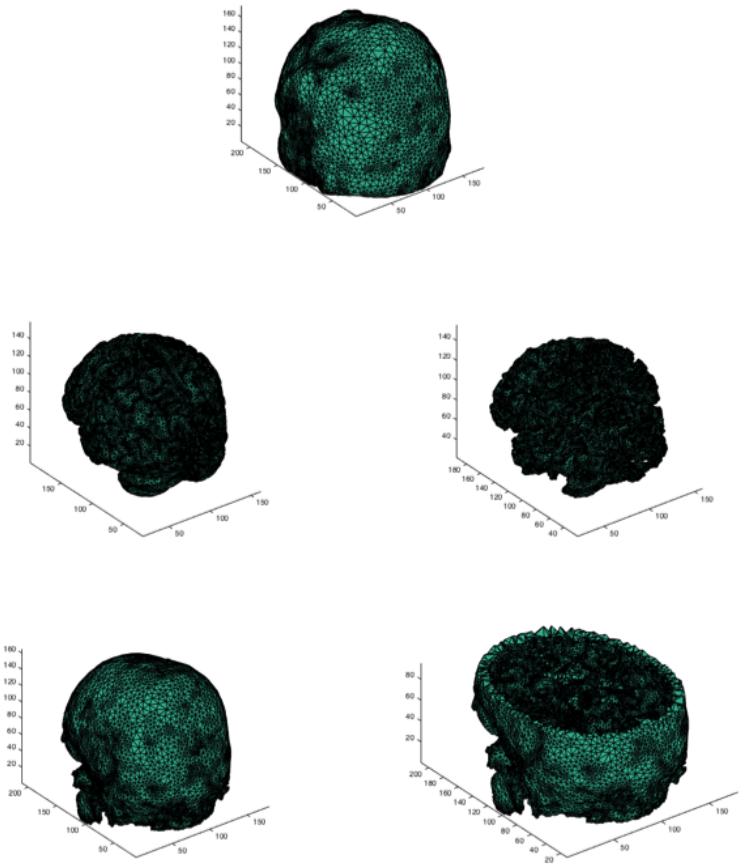


Figure 6.16: Pipeline 1 – The four regions shown separately, in the following order: SCALP/SKULL, GM, WM, CSF – overview, CSF – section view.

the user can choose the lower and/or upper boundary and the increase/decrease slope (Fig. 6.11). Unfortunately, the images are not perfect and there is no clear definition of a threshold for every domain. A certain intensity may correspond to several domains. The decision is left to the intuition of the user. Table 6.2 contains the values of the program-specific indicators we set for every domain.

Segmentation

We identified the domains in the following order: SK (skull), CSF, GM, WM. The anatomical model resulted is shown in Fig. 6.12. In Fig. 6.13 a single layer is depicted,

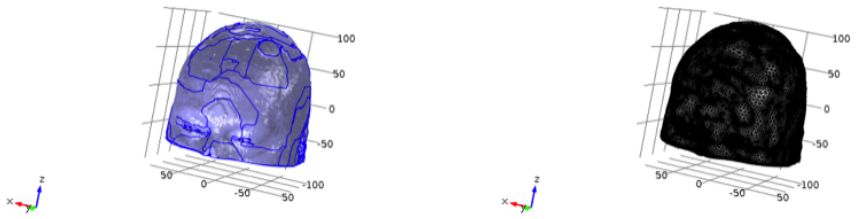


Figure 6.17: Pipeline 1 – The skull (SK) domain: a) geometry imported in COMSOL; b) mesh generated in COMSOL from the previously imported geometry.

the model shows empty spaces, meaning regions that are not allocated to any domain.

To Matlab (*.img*, Analyze format)

We created the FEM model with iso2mesh [248], [249], an open-source package for Matlab/Octave for mesh generation and processing. With iso2mesh, 3D FEM meshes can be created from surfaces, 3D binary files (as this case) and segmented volumetric images (for the latter case, you cannot use the *cgalmesh* method, details below).

As options of *cgalmesh* – considered the most robust function in the documentation – the maximum size of a triangle has 15 pixels and the maximum volume of a tetrahedron is 100 voxels. The function *v2m* created 871,468 triangles and 1,056,979 tetrahedra from 202,158 nodes.

In Fig. 6.14 the mesh nodes are displayed in three dimensions and in Fig. 6.15 is shown the 3D mesh with the four identified regions – overview and section on Oz axis. The four regions and a section through CSF are represented separately in Fig. 6.16.

To COMSOL (*.stl*, *.vrml*)

Geometries and meshes (*.stl*, *.vrml* v1 and *.vtu* (VTK format) can be imported in COMSOL. If the files with these extensions represent 3D volumes, they can be imported into COMSOL as geometries and then used to generate volumetric meshes for a single domain. If the files represent meshes, they should be imported directly as meshes. Whether as geometry or mesh, just a few file formats are supported.

We were able to import in COMSOL only the SK (skull/scalp) domain as a geometry (2,293 boundaries, 8,954 edges, 6,619 vertices), which is shown in Fig. 6.17a.

It was then possible to generate the mesh associated with the geometry imported (6,619 vertex elements, 52,130 edge elements, 308,484 boundary elements), the mesh is shown in Fig. 6.17b.

The other three domains could not be imported in COMSOL as geometries (errors in COMSOL: *"Face cannot be parametrized"* or *"Failed to analyze local face topology"*). This might be due to the faults in the segmented (CAD) model or the values of the program-specific indicators.

Pipeline 2 (start from the Colin27 CAD model) – Fig. 6.10b

An already segmented CAD model is available online [250] as a volumetric mesh model,

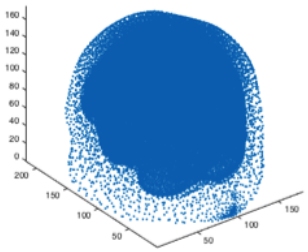


Figure 6.18: Pipeline 2 – The nodes (vertices) in the 3D mesh after segmentation (Colin27).

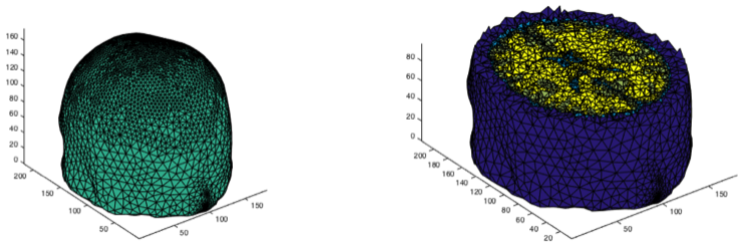


Figure 6.19: Pipeline 2 – The 3D mesh with the four segmented regions colored differently – overview and section view (Colin27).

created from the Colin27 MRI atlas [251]; the source atlas consists of MR images of the same individual, scanned 27 times; the geometrical model data is stored in a .mat file representing a mesh.

After eliminating some of the options (the use of proprietary programs such as Mimics from Materialise and ScanIP from Simpleware, both accepted by COMSOL [252], [253], finding programs that generate geometries/meshes from *.mat* geometries, creating a model in COMSOL from Matlab through LiveLink – it can be used only for creating a geometry from scratch), we have found a package for Matlab that processes *.stl* [254] (reading, writing in binary/ascii format, minor processing, visualizing).

The Colin27 model (*.mat*) consists of the following data structures:
node: a matrix with 3 columns representing the node coordinates (in mm);
face: describes the triangles; a matrix with 4 columns, 3 columns containing the indices of the nodes forming a triangle and the last column representing the domain index as follows: 1-scalp/skull, 2-CSF, 3-gray matter, 4-white matter;
elem: describes the tetrahedra; a matrix with 5 columns, 4 columns containing the indices of the nodes forming a tetrahedron and the last column representing the domain index as follows: 1-scalp/skull, 2-CSF, 3-gray matter, 4-white matter.

The Colin27 mesh (which includes the scalp/skull domain) presents obvious differences from the mesh previously segmented in Pipeline 1. The Colin27 mesh is compact; it does not have "holes" or other "defects". This is due to the techniques for accuracy improvement previously applied to the Colin27 mesh [250]. The representations are in Fig. 6.18, 6.19 and 6.20 whereas the details of the mesh are specified in [250]: 70,226 nodes, 423,375

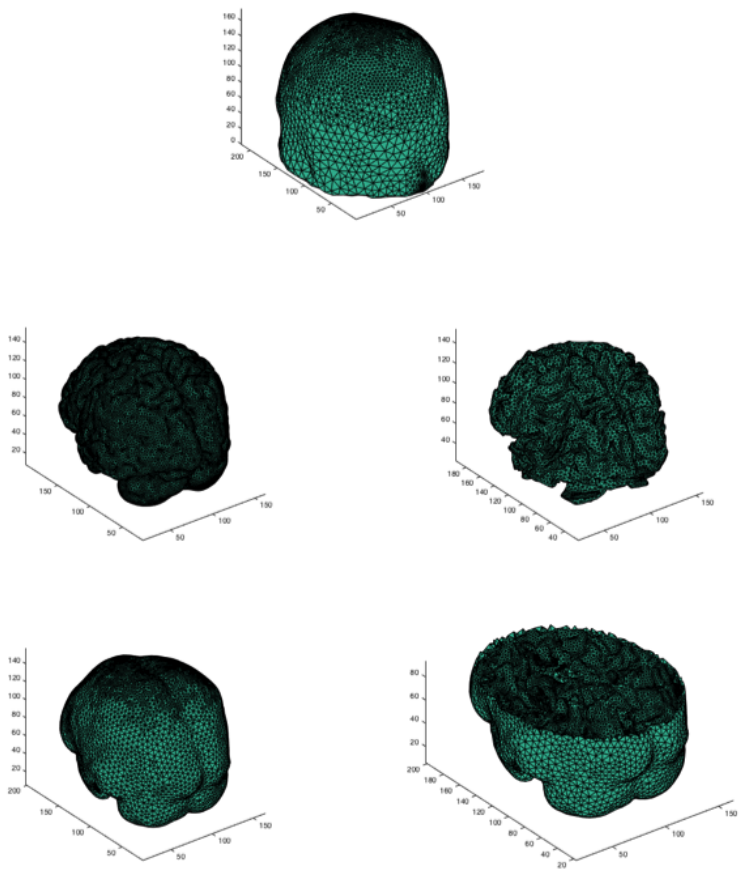


Figure 6.20: Pipeline 2 – The four regions shown separately, in the following order: SCALP/SKULL, GM, WM, CSF – overview, CSF – section view (Colin27).

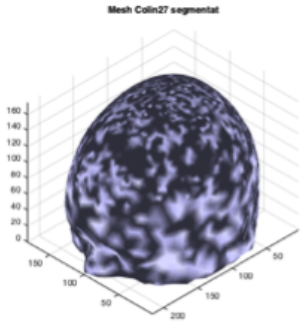


Figure 6.21: Pipeline 2 – The Colin27 model, nodes and faces.

tetrahedra and 119,554 triangles (6,916 for the scalp surface, 12,780 for CSF including ventricles, 61,026 for the GM surface and 38,832 for WM). Note that the number of tetrahedra is close to the mesh generated in Pipeline 1, but the latter has a much larger number of triangles, probably because of the "holes" in the 3D volume so that many triangles do not get shared by the tetrahedra.

The whole mesh (nodes and faces) is represented in Fig 6.21.

After writing four STL files corresponding to the four domains, we tried to import

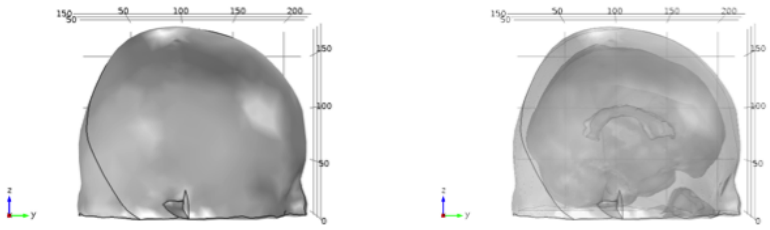


Figure 6.22: Pipeline 2 – The Colin27 model, geometry import into COMSOL, SK and CSF (zOy plane).

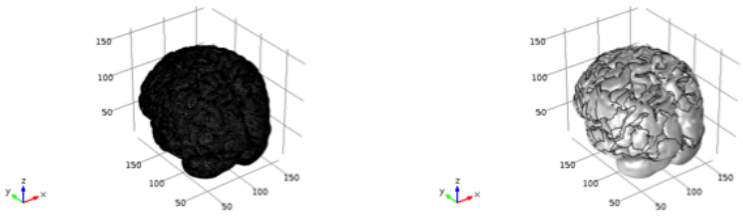


Figure 6.23: Pipeline 2 – The Colin27 model, GM domain, imported mesh and the associated geometry.

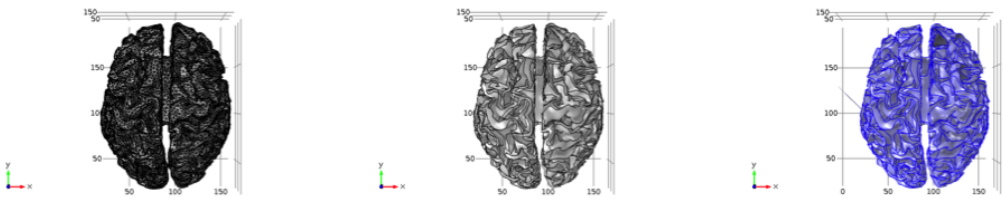


Figure 6.24: Pipeline 2 – The Colin27 model, WM domain, imported mesh, associated geometry and generated geometry from mesh (yOx plane).

them as geometry into COMSOL. We have succeeded this only for SK and CSF (Import Geometry, without the Simplify mesh checkbox), shown in Fig. 6.22.

Given that even for this very "clean" mesh the import of geometries into COMSOL was not successful for all the domains, we decided to import the models as meshes into COMSOL. COMSOL starts from the coordinates of the nodes and triangles and interpolates to build the continuous 3D object. The *"Create Geometry from Mesh"* option only succeeded for WM, but for GM we can see the geometry associated with the imported mesh. In WM, the associated and generated geometry seem identical. Fig. 6.23 and 6.24 show the GM and WM domains, geometry and mesh.

6.2.2 TMS for realistic models

The realistic geometrical model extracted in Section 6.2.1 can replace the sphere used in Section 6.1 to represent the human head. We solved the TMS problem described in Section 6.1 with FEM (in COMSOL), in order to validate the TMS procedure and to extend it for the realistic geometrical models developed in Section 6.2.1.

We developed three FEM numerical models for the first test case (with one circular

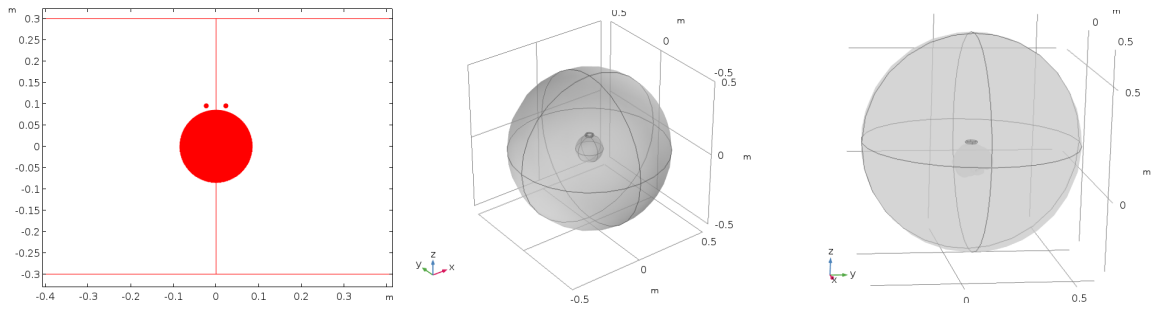


Figure 6.25: The three geometric models in COMSOL.

coil above the head):

- an axysimmetrical 2.5D model, describing one circular coil placed above a sphere – called *sphere2.5Daxi* (Fig. 6.25a)
- a 3D model, with one circular coil placed above a sphere – called *sphere3D* (Fig. 6.25b)
- a 3D model, with one circular coil placed above a realistic geometrical model (the CSF domain from Fig. 6.22) – called *real3D* (Fig. 6.25c)

The coil (modeled as a torus) and the sphere domain (for models 1 and 2) have the geometrical and material properties in Section 6.1.1 (with angle $\beta = 0$). The two domains (coil and head) are surrounded by a domain of air, modeled as a cylinder for model 1 and as a sphere for models 2 and 3, with the radius of 0.5 m.

The equations are those describing a MQS field, with magnetic insulation on the boundary of the air domain and gauge fixing for the **A**-field. The coil carries a current with a varying rate of 25 MA/s, as in Section 6.1.1. The coil and air domains should have null conductivity, but in order to avoid numerical instabilities, we use a small value for $\sigma = 1$ S/m. The simulation of 10 μs took on a 1.4GHz Intel Core i5, RAM 4 GB 8 seconds for model 1, 15 minutes for model 2, and 8 minutes for model 3, the differences being due to the different meshes needed for their solving, described below.

The tetrahedral mesh uses the predefined size for the elements, consisting of:

- 1,666 domain elements and 129 boundary elements (coarser element size for all three domains) for the *sphere2.5Daxi* model, resulting in 3,410 DoFs.
- 13,066 domain elements, 1,214 boundary elements, and 220 edge elements (normal element size for air domain, fine element size for coil and head-sphere) for the *sphere3D* model, resulting in 103,273 DoFs.
- 17,084 domain elements, 1,676 boundary elements, and 153 edge elements (normal element size for all three domains) for the *real3D* model, resulting in 134,115 DoFs.

The induced electric field in the head for the three FEM models is represented in Fig. 6.26, whereas in Fig. 6.27 is shown the tangential component of the induced field on an upper right quarter of a meridian line for the *sphere2.5Daxi* model (left), on the upper half of a meridian line for the *sphere3D* model (middle) and on the whole head's surface for the *real3D* model (right). Fig. 6.26 reproduces the induced electric field in the head

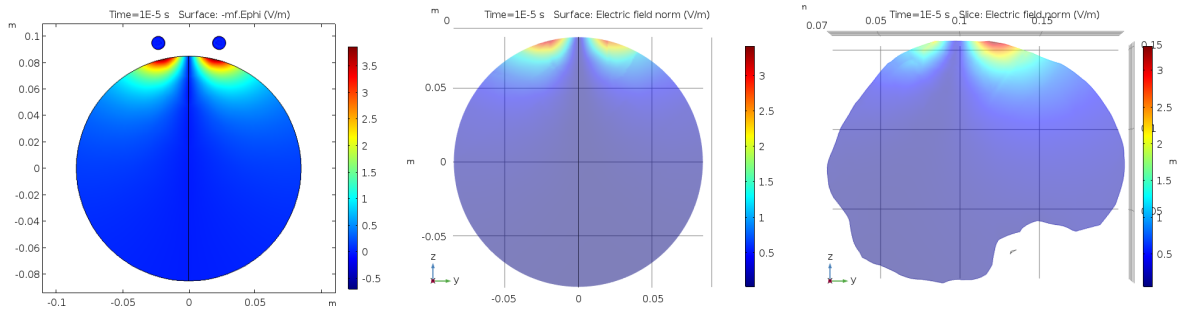


Figure 6.26: Induced field intensity for the three models. The figures are to be compared with Fig. 6.5 in Section 6.1.

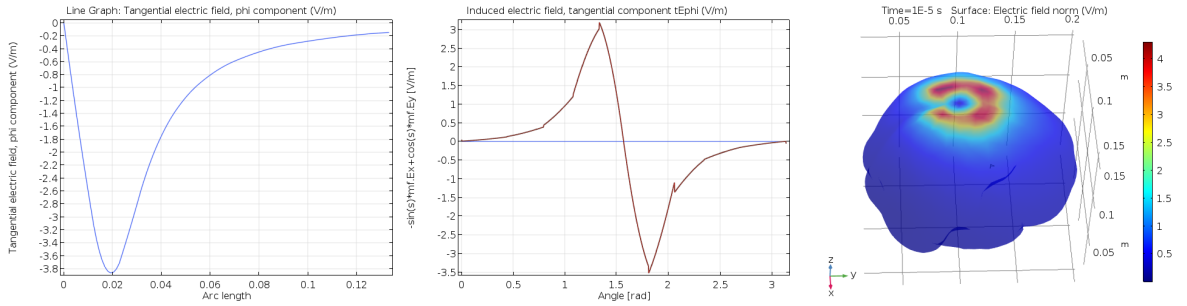


Figure 6.27: The tangential component of the induced electric field on the head's surface: left: on an upper right quarter of a meridian line; middle: on the upper half of a meridian line; right: on the whole head's surface. The figures are to be compared with A1 from Fig. 6.6 in Section 6.1.

(to be compared with Fig. 6.5) and Fig. 6.27 reproduces A1 from Fig. 6.6. These results validate qualitatively and quantitatively both the numerical model in Section 6.1 and the FEM model, in particular the FEM model for the realistic geometries extracted in Section 6.2.1. The differences in the values are due to the different meshes needed for the proper solving of every model.

6.3 Conclusion regarding TMS

This chapter proposes a procedure based on MG+EC(t) field to extract a hierarchy of models of varying accuracy for TMS, which can be further used for efficient optimization of the TMS device. These models can be successfully used as surrogate models for optimization of customized TMS systems with accuracy control. This approach is one of the most important research topics in CSE, as is noted in [2]: *"recent years have seen increasing recognition of the critical role of uncertainty quantification (UQ) in all phases of the CSE lifecycle, from inference to prediction, to optimization [...] the traditional approach to theory in numerical analysis provides only an insufficient basis to quantify the efficiency of algorithms and software, since many theorems are only qualitative and leave the constants unspecified"*.

We deducted precise expressions for the upper bound of the relative numerical errors

(6.18), (6.20), resulting in the upper bound of the global relative error:

$$\varepsilon < (c_1/n_t)^{p_1} + c_3/M, \text{ where } c_1 = 4.2,\ c_3 = 15,\ p_1 = 8,\ M = n_u \times n_v \tag{6.21}$$

The algorithms complexities are for MG: $O\left(n_t \cdot n_u \cdot n_v\right)$; for EC: $O\left(n_y \cdot n_u \cdot n_v\right)^2$; for EC-BEM: $O\left(n_u \cdot n_v\right)^2$ and reduced to a linearithmic one by FFT-FMM.

This dependence allows the identification of the necessary level of refinement for an acceptable tolerance of the model. The reference model is simple, but it does not neglect the essential aspects of TMS. The head is modeled as a conductive sphere and the drive coil is a wire of arbitrary shape. The relatively small time variation of the coil current allows the calculation of field induced by a MG+EC(t) regimes sequence, coupled by Faraday's law of induction. This approach, which neglects the magnetic effects of eddy currents, is more effective than considering the MQS regime. The magnetic field produced by the excitation coil does not depend on what happens inside the head, and is calculated by Biot-Savart-Laplace integral, which is estimated by numerical quadrature.

The accuracy study performed enables the control of approximation error. The proposed method is superior to the method of approximating the wire by a polygon [211]. The result of numerical integration is then used to determine the induced field, which has two components: a solenoidal component determined by the time derivative of magnetic vector potential and a irrotational component, expressed with a scalar potential. In the case of homogenous conductor the scalar potential is a harmonic function, solution of the Laplace equation, which has an analytical expression, obtained by Fourier series expansion of spherical harmonics. In the axisymmetric case, this secondary component does not exist, because the magnetic potential is tangential and it does not have normal component. In the nonsymmetrical cases studied, such as the coil with three sections, this component is about 5 times smaller than the primary one. Another important aspect found is that the double layer potential is about 40 times smaller than the single layer one, which leads to a rapid solving method for the iterative BEM equation. The proposed MG+EC(t) approach reduces the problem of field computing to two numerical integrations: one 1D along the coil's wire, and other 2D, over the head surface.

A pipeline for the generation of a realistic 3D geometrical model for the human head from medical imaging data (EEG) is then proposed. The procedure has two options, either involving segmentation of EEG images using dedicated software, or by starting from a segmented model. Whatever option is used, the resulted CAD model can subsequently be used in simulation software such as Matlab or Comsol.

The approaches proposed for TMS modeling in the first part of the chapter are customized for the realistic modeling of TMS, with the more complex geometry of the 3D CAD model extracted from EEG. The modeling can become more realistic when the conductive domain is heterogeneous and anisotropic, this approach being suitable for personalized treatment. The analysis of these complicated cases can be performed by perturbation of the model of the head and it has advantages over direct solving of the complicated problems.

Chapter 7

Conclusions

This chapter presents the original contributions of the research underlying this thesis, the papers in which the results were disseminated and future research topics.

7.1 Original contributions

1. Critical discussion on the state of the art in multiscale modeling of biological systems in general and neuronal systems in particular, stressing on the importance of model reduction within this procedure.
2. Study of a 1D-EQS transmission line model for the neuronal axon. Extraction of approximate analytical expressions and an innovative numerical procedure for the computation of the characteristic parameters of the 1D-EQS linear model: the nerve conduction velocity, attenuation and maximum transmission length, based on an original definition of the delay time. Estimation of upper and lower boundaries of the transmission velocity. Analytical and numerical prediction of the dependence velocity vs. axon diameter in myelinated and unmyelinated axons. Realistic numerical simulation by digitization of the standard neuronal signal.
3. Development of a hierarchical series of 9 types of models for myelinated compartments, corresponding to three spatial geometry classes (2.5D, 1D and 0D) and three computational approaches for each geometry (analytical, numerical and reduced order models). These models are hierarchized based on modeling errors, which are closely related to the complexity of the models. Introduction of an error estimator suitable to neural signals, based on a weighted norm, in order to correctly assess the models' accuracy.
4. A procedure for the efficient extraction of a reduced model of a myelinated axon: reduced models of myelinated compartments are connected with models of Ranvier nodes and the global model thus obtained is again reduced. The extracted model is able to reproduce the saltatory conduction with controlled accuracy.
5. Extraction of a model of Transcranial Magnetic Stimulation (TMS) from the field equations MG+EC(t); the human head is modeled as a homogenous conductive

sphere and the magnetic vector potential is computed by approximating the Biot-Savart-Laplace integral with numerical quadrature. Introduction of a BEM procedure to extract a hierarchy of models of varying accuracy for TMS, by deducting precise expressions for the upper bound of the relative numerical errors, leading to the upper bound of the global relative error.

6. Two efficient pipelines for the generation of a 3D geometrical CAD model from medical imaging data (MR/CT images). Extension of the proposed approach for TMS modeling to the geometrically realistic 3D model of the human head.

Most of the results presented here are published in the papers listed in Section 7.3 (the author of this thesis being first author for 7 of them).

7.2 Future research

1. Extraction of the analytical solution for the EQS 2.5D problem formulated in Chapter 4 and the numerical solution with BEM, based on the Green function previously extracted.
2. Modeling the extracellular 3D space of axons and large-scale neural circuits in the brain.
3. Study the effects of pathologies such as demyelination on the normal functioning of the neural system, due to the impact on the transmission parameters (attenuation, velocity, maximum transmission length, delay time).
4. Extension of TMS modeling approach to heterogeneous and anisotropic conductive domain, which is more suitable for personalized treatment.
5. Validation of the segmented model from Pipeline 1 with the model Colin27 as reference, based on similarity indicators such as Dice coefficient or Tanimoto coefficient.

7.3 Dissemination of results

1. **Ruxandra Bărbulescu**, Aurel-Sorin Lup, Gabriela Ciuprina, Daniel Ioan and A. Egemen Yilmaz, "Intelligent Particle Swarm Optimization of Superconducting Magnetic Energy Storage devices", International Symposium on Fundamentals of Electrical Engineering 2014, ISFEE2014, NOVEMBER 28-29, 2014, Bucharest, Romania, pg. 1-4, DOI: 10.1109 / ISFEE.2014.7050607.
2. **Ruxandra Bărbulescu** and Daniel Ioan. "Multi-scale and reduced modelling of biological systems.", 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE), 2015, pp. 597-602. IEEE, 2015, DOI: 10.1109 / ATEE.2015.7133876. Developed in section 2.1.
3. Mihai Popescu, Aurel-Sorin Lup, **Ruxandra Bărbulescu**, Gabriela Ciuprina and Daniel Ioan, "Using Object Oriented Data Structures for Optimizing MEMS Devices

on Parallel Computers", 17th International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering, 2015, pp. 1-4, ISBN: 978-84-606-9102-0.

4. **Ruxandra Bărbulescu**, "Simulation of electric potential in 3D neuronal axon", IEEE International Symposium on Fundamentals of Electrical Engineering 2016, ISFEE2016, June 30 – July 2, ISBN: 978-1-4673-9574-8, DOI: 10.1109 / ISFEE.2016.7803206, WOS: 000392434400058.

5. Gabriela Ciuprina, Daniel Ioan, Aurel-Sorin Lup, Mihai Popescu, **Ruxandra Bărbulescu** and Alexandra Ştefănescu, "Coupled multiphysics-RF reduced models for MEMS", 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016, July 04–06, at Delhi Technological University, Delhi, India. DOI:10.1109 / ICPEICES.2016.7853254.

6. **Ruxandra Bărbulescu**, Daniel Ioan and Jean Ciurea, "Simple 1D models for neuro-signals transmission along axons", IEEE International Conference and Exposition on Electrical and Power Engineering (EPE), 20-22 Oct. 2016, Iasi, Romania, pp. 313-319, ISBN: 978-1-5090-6128-0, DOI:10.1109 / ICEPE.2016.7781354, WOS:000390706300064, ISSN: 2471-6855. Developed in Chapter 3.

7. Daniel Ioan, **Ruxandra Bărbulescu**, Jean Ciurea, Gabriela Ciuprina, and Przemyslaw Syrek. "Simple hierarchical models of the Transcranial Magnetic Stimulation.", 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), 23-25 March 2017, pp. 398-403. IEEE, ISBN: 978-1-5090-5160-1, DOI: 10.1109 / ATEE.2017.7905129. Developed in Chapter 6.

8. Przemyslaw Syrek, **Ruxandra Bărbulescu**, "Parametric curves to trace the TMS coils windings", 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), 23-25 March 2017, pp. 386-391, IEEE, ISBN:978-1-5090-5160-1, DOI: 10.1109 / ATEE.2017.7905069.

9. **Ruxandra Bărbulescu**, Daniel Ioan, Gabriela Ciuprina, "Modeling the Saltatory Conduction in Myelinated Axons by Order Reduction", 20th International Conference on Neuroinformatics and Computational Neuroscience, ICNCN 2018, August 27-28. Developed in Chapter 5.

10. **Ruxandra Bărbulescu**, Daniel Ioan, Gabriela Ciuprina, Aurel Sorin Lup, and Mihai Popescu, "Reduced Order Models for the Simulation of the Saltatory Conduction", The 12th International Conference on Scientific Computing in Electrical Engineering, SCEE 2018, September 23-27, 2018 – Taormina, Sicily, Italy.

11. Daniel Ioan, **Ruxandra Bărbulescu**, Luis Miguel Silveira, Gabriela Ciuprina, "Reduced Order Models of Myelinated Axonal Compartments", under review. Developed in Chapter 4.

12. **Ruxandra Bărbulescu**, Daniel Ioan, Gabriela Ciuprina, "Coupled Macromodels for the Simulation of the Saltatory Conduction", Scientific Bulletin – University Politehnica of Bucharest, Series C, under review.

Appendix

A1. 2.5D – EQS model FIT code

```
clear all;
close all hidden;
clc;
% prepare path
restoredefaultpath;
sourcespath = genpath(pwd);
addpath(sourcespath);

debug = 'off'; %'off'; %'on'; % if 'on' it draws all sorts of figures
tic
\% tests
flag = 'LMicECEc';  % 'LMicECEc' / 'LMicNonECEc', 'LMicAll';

if (1==1)
    vector_pas = [5e-7];
    idx1 = 1;
    idx2 = 2;
end

no_pasi = length(vector_pas);
nr_noduri = zeros(size(vector_pas));

for idx_pas = 1:no_pasi
    pas = vector_pas(idx_pas);
    [teste,pasnou,nrNoduri] = citeste_teste_axonEQS(flag,pas);
    nr_teste = length(teste);

    for idx_teste = 1:nr_teste
        fprintf('--> test %d: \n',idx_teste);
        model = teste{idx_teste};
        model.term = find_idx_terminals_axon(model); % terminals info with respect to nodes index
        model.figno = idx_teste*10;
        model.draw = 'off';

        if strcmpi(debug,'on')
            figure(model.figno); clf;
            draw_domain_axon(model.geom);
            hold on;
            draw_grid_axon(model.geom,model.grid);
            draw_terminals_axon(model.geom,model.term);
            %number_nodes_axon(model.geom,model.grid);
            %show_nodes_terminals_axon(model);
        end

        [out,Vterm1,Vterm2] = solve_EQS_axon_MsiC_omogene_izotrope(model);

        %% for Richardson extrapolation
%         Vterm11{idx_pas}=Vterm1(:,1);
%         Vterm12{idx_pas}=Vterm2(:,1);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end

%% Richardson extrapolation
% if (length(Vterm11{1})*2-1 == length(Vterm11{2}))
%     for kk=1:length(Vterm11{1})
%         matrR11(1,kk) = Vterm11{1}(kk);
%         matrR11(2,kk) = Vterm11{2}(2*kk-1);
%         matrR12(1,kk) = Vterm12{1}(kk);
%         matrR12(2,kk) = Vterm12{2}(2*kk-1);
%     end
%     extRe = extrapolareRichardson( vector_pas, real(matrR11), real(matrR12) )
%     extIm = extrapolareRichardson( vector_pas, imag(matrR11), imag(matrR12) )
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
toc


function [teste,pasnou,nrNoduri] = citeste_teste_axonEQS(flag,pas)
```

```
idx_test = 1;

a = 7e-6;       %m
b = 10e-6;      %m
sigC = 1.0823;                  % S/m - cytoplasm
sigR = 2.04e-4;                 % S/m - membrane (may be anisotropic)
sigZ = 2.04e-4;                 % S/m - membrane (may be anisotropic)
epsilonVID = 8.854187817e-12;   % F/m
epsrM = 15.44;

r = 1 / (sigC*pi*a^2);
g = (sigR*2*pi)/log((b-a)/a+1);
% c = (epsM*2*pi)/log(a/b+1);

l0 = 1/sqrt(r*g);
lmic = 0.25*l0;
lmediu = l0;
lop = 2.5*l0;
lmare = 4*l0;
loptim = 9*l0;

ltest = lmic;

%x0 = 0;    % stick to axis
x0 = eps;   % model with a small hole in the middle

[nx1,nx2,nz,pasnou] = get_noduri_mesh(pas,a,b-a,ltest);
nrNoduri = (nx1+nx2-1)*nz;

fmin = 1e3;     % Hz
fmax = 1e7;     % Hz
nop = 10;   % number of frequency points

if or(strcmpi(flag,'LMicECEv'),strcmpi(flag,'LMicAll'))

    %% test 1 axon, small length, sparse grid, excitation ECE in voltage

    model.frecv.fmin = fmin;
    model.frecv.fmax = fmax;
    model.frecv.nop = nop;
    model.frecv.filename = 'LMicECEv';  % filename root
    model.frecv.folder = 'out';         % folder name (relative to main program folder) for frequency
        responses

    % geometry
    model.geom.a = a; % m
    model.geom.b = b; % m

    model.geom.l = ltest;
    model.geom.x0 = x0;
    model.geom.z0 = 0;
    %model.geom.sim = 'xy'; % 'xy' or 'axi' (rotation axis is Oz)
    model.geom.sim = 'axi';

    model.matC.sig = sigC;

    model.matM.sigR = sigR;
    model.matM.sigZ = sigZ;
    model.matM.eps = epsrM * epsilonVID;

    % terminals: there has to be one 'g' - ground;
    % type of excitation: ECEv, ECEc, nonECEc

    tipExc = 'ECEv'; %'nonECEc'; %'ECEv';

    model.term{1}.exc = 'g'; % ground
    model.term{1}.poz = 'xmax'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{1}.min = model.geom.z0;
    model.term{1}.max = model.geom.z0 + model.geom.l;
    %
    model.term{2}.exc = tipExc;
    model.term{2}.poz = 'zmax'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{2}.min = model.geom.x0;
    model.term{2}.max = model.geom.x0 + model.geom.a;
    %
    model.term{3}.exc = tipExc;
    model.term{3}.poz = 'zmin'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{3}.min = model.geom.x0;
    model.term{3}.max = model.geom.x0 + model.geom.a;
    %

    % grid
    model.grid.nx1 = nx1; % discretization on [0,a]
    model.grid.nx2 = nx2; % discretization on [a,b]
    model.grid.nz = nz;
    model.info = strcat('l = ',num2str(model.geom.l),'nx1 = ',num2str(model.grid.nx1),...
        'nx2 = ',num2str(model.grid.nx2),...
        'nz = ',num2str(model.grid.nz));
    model.frecv.filename = strcat(model.frecv.filename,'_',num2str(model.grid.nx1),'_',num2str(model.
        grid.nx2),'_',num2str(model.grid.nz));

    teste{idx_test} = model;
    idx_test = idx_test + 1;
```

```
end

if or(strcmpi(flag,'LMicECEc'),strcmpi(flag,'LMicAll'))

    %% test 1 axon, small length, sparse grid, excitation ECE in current
    model.frecv.fmin = fmin;
    model.frecv.fmax = fmax;
    model.frecv.nop = nop;
    model.frecv.filename = 'LMicECEc';  % filename root
    model.frecv.folder = 'out';         % folder name (relative to main program folder) for frequency
        responses

    % geometrie
    model.geom.a = a; %m
    model.geom.b = b; %m
    model.geom.l = ltest;
    model.geom.x0 = x0;
    model.geom.z0 = 0;
    %model.geom.sim = 'xy'; % 'xy' sau 'axi' (rotation axis is Oz)
    model.geom.sim = 'axi';

    model.matC.sig = sigC;

    model.matM.sigR = sigR;
    model.matM.sigZ = sigZ;
    model.matM.eps = epsrM * epsilonVID;

    % terminals: there has to be one 'g' - ground;
    % type of excitation: ECEv, ECEc, nonECEc

    tipExc = 'ECEc'; %'nonECEc'; %'ECEv';

    model.term{1}.exc = 'g'; % ground
    model.term{1}.poz = 'xmax'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{1}.min = model.geom.z0;
    model.term{1}.max = model.geom.z0 + model.geom.l;
    %
    model.term{2}.exc = tipExc;
    model.term{2}.poz = 'zmax'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{2}.min = model.geom.x0;
    model.term{2}.max = model.geom.x0 + model.geom.a;
    %
    model.term{3}.exc = tipExc;
    model.term{3}.poz = 'zmin'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{3}.min = model.geom.x0;
    model.term{3}.max = model.geom.x0 + model.geom.a;
    %

    % grid
    model.grid.nx1 = nx1; % discretization on [0,a]
    model.grid.nx2 = nx2; % discretization on [a,b]
    model.grid.nz = nz;
    model.info = strcat('l = ',num2str(model.geom.l),'nx1 = ',num2str(model.grid.nx1),...
        'nx2 = ',num2str(model.grid.nx2),...
        'nz = ',num2str(model.grid.nz));
    model.frecv.filename = strcat(model.frecv.filename,'_',num2str(model.grid.nx1),'_',num2str(model.
        grid.nx2),'_',num2str(model.grid.nz));

    teste{idx_test} = model;
    idx_test = idx_test + 1;
end

if or(strcmpi(flag,'LMicNonECEc'),strcmpi(flag,'LMicAll'))

    %% test 1 axon, small length, sparse grid, excitation nonECE in current
    model.frecv.fmin = fmin;
    model.frecv.fmax = fmax;
    model.frecv.nop = nop;
    model.frecv.filename = 'LMicECEc';  % filename root
    model.frecv.folder = 'out';         % folder name (relative to main program folder) for frequency
        responses

    % geometrie
    model.geom.a = a; %m
    model.geom.b = b; %m
    model.geom.l = ltest;
    model.geom.x0 = x0;
    model.geom.z0 = 0;
    %model.geom.sim = 'xy'; % 'xy' sau 'axi' (rotation axis is Oz)
    model.geom.sim = 'axi';

    model.matC.sig = sigC;

    model.matM.sigR = sigR;
    model.matM.sigZ = sigZ;
    model.matM.eps = epsrM * epsilonVID;

    % terminals: there has to be one 'g' - ground;
    % type of excitation: ECEv, ECEc, nonECEc

    tipExc = 'nonECEc'; %'nonECEc'; %'ECEv';

    model.term{1}.exc = 'g'; % ground
    model.term{1}.poz = 'xmax'; % 'xmax', 'zmin', 'xmax', 'zmax'
```

```
    model.term{1}.min = model.geom.z0;
    model.term{1}.max = model.geom.z0 + model.geom.l;
    %
    model.term{2}.exc = tipExc;
    model.term{2}.poz = 'zmax'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{2}.min = model.geom.x0;
    model.term{2}.max = model.geom.x0 + model.geom.a;
    %
    model.term{3}.exc = tipExc;
    model.term{3}.poz = 'zmin'; % 'xmax', 'zmin', 'xmax', 'zmax'
    model.term{3}.min = model.geom.x0;
    model.term{3}.max = model.geom.x0 + model.geom.a;
    %

    % grid
    model.grid.nx1 = nx1; % discretization on [0,a]
    model.grid.nx2 = nx2; % discretization on [a,b]
    model.grid.nz = nz;

    model.frecv.filename = strcat(model.frecv.filename,'_',num2str(model.grid.nx1),'_',num2str(model.
        grid.nx2),'_',num2str(model.grid.nz));

    teste{idx_test} = model;
    idx_test = idx_test + 1;
end


function [ out ] = extrapolareRichardson( pasi, matrR11, matrR12 )
%% Richardson extrapolation
%  steps: lxnr_pasi, contains the values of the steps used for
%         extrapolation
%  matrR11: matrix length(pasi) x nr_puncte: R11 at step 1,2,..
%  matrR12: matrix length(pasi) x nr_puncte: R12 at step 1,2,..
    h1 = pasi(1);
    h2 = pasi(2);
    h1p = h1^2;
    h2p = h2^2;
    dif = h2p-h1p;

    for i=1:length(matrR11)
        R11(1) = matrR11(1,i);
        R11(2) = matrR11(2,i);
        R12(1) = matrR12(1,i);
        R12(2) = matrR12(2,i);

        R11extrap = (R11(1)*h2p-R11(2)*h1p)/dif;
        R12extrap = (R12(1)*h2p-R12(2)*h1p)/dif;

        Rextrap = [R11extrap R12extrap; R12extrap R11extrap]
        Gextrap = inv(Rextrap);
        out.R11extrap(i) = Rextrap(1,1);
        out.R12extrap(i) = Rextrap(1,2);
        out.G11extrap(i) = Gextrap(1,1);
        out.G12extrap(i) = Gextrap(1,2);
    end
end
```

Functions for solving

```
function term = find_idx_terminals_axon(model)
    geom = model.geom;
    term = model.term;
    grid = model.grid;

    xs = geom.x0;
    xd1 = geom.x0 + geom.a;
    xd2 = geom.x0 + geom.b;
    zs = geom.z0;
    zd = geom.z0 + geom.l;
    nx1 = grid.nx1;
    nx2 = grid.nx2;
    nz = grid.nz;

    grid_x1 = linspace(xs,xd1,nx1);
    grid_x2 = linspace(xd1,xd2,nx2);
    grid_z = linspace(zs,zd,nz);
    grid_x = [grid_x1 grid_x2(2:end)];

    nx = nx1 + nx2 - 1;

    not  = length(term); % no of terminals
    for i = 1:not
        vmin = term{i}.min;
        vmax = term{i}.max;
        switch term{i}.poz
            case 'xmin'
                idx_j = find(and(grid_z>=vmin,grid_z<=vmax));
                term{i}.idx = idx_j;
            case 'xmax'
                idx_j = find(and(grid_z>=vmin,grid_z<=vmax));
                term{i}.idx = (nx-1)*nz + idx_j;
            case 'zmin'
```

```
                idx_i = find(and(grid_x>=vmin,grid_x<=vmax));
                term{i}.idx = (idx_i-1)*nz + 1;
            case 'zmax'
                idx_i = find(and(grid_x>=vmin,grid_x<=vmax));
                term{i}.idx = (idx_i-1)*nz + nz;
    %         case 'zmijloc'
    %             idx_i = find(and(grid_x>=vmin,grid_x<=vmax));
    %             term{i}.idx = (idx_i-1)*nz + round(nz/2);
            otherwise
                error('unknown pozition of terminal');
        end
    end
end


function [nxa,nxb,nz,pasnou] = get_noduri_mesh(pas,a,b,z)

    nrinta = [floor(a/pas) round(a/pas)];
    pasao = a./nrinta;
    [pasa,idxinta] = min(pasao);
    nxa = nrinta(idxinta)+1;

    nrintb = [floor(b/pas) round(b/pas)];
    pasbo = b./nrintb;
    [pasb,idxintb] = min(pasbo);
    nxb = nrintb(idxintb)+1;

    nrintz = [floor(z/pas) round(z/pas)];
    paszo = z./nrintz;
    [pasz,idxintz] = min(paszo);
    nz = nrintz(idxintz)+1;

    pasi = [pasa pasb pasz];
    pasnou = mean(pasi);
end


function A = MatriceIncidentaLaturiNoduri(grid)

nx = grid.nx;
nz = grid.nz;

N = nx*nz;
noRows = N + nz;
noCols = 2*N;

A = sparse(noRows,noCols);

% Ox - towards E
% Oz - towards N
for k = 1:N     % browse the nodes
    doik = 2*k;
    doikm1 = doik-1;
    A(k,doik) = 1;          % current node, edge towards N
    A(k,doikm1) = 1;        % current node, edge towards E
    A(k+1,doik) = -1;       % node N, edge towards N
    A(k+nz,doikm1) = -1;    % node E, edge towards E
end
% correct false nodes, N border
for i = 1:nx
    idx_nod = nz*i;
    idx_lat = 2*idx_nod;
    A(idx_nod,idx_lat) = 0;   % was 1
    A(idx_nod+1,idx_lat) = 0; % was -1
end
% correct false nodes, E border
for j = 1:nz
    idx_nod = (nx-1)*nz + j;
    idx_lat = 2*idx_nod - 1;
    A(idx_nod,idx_lat) = 0;     % was 1
    A(idx_nod+nz,idx_lat) = 0; % was -1
end
% erase unnecessary nodes from the right of E border
A(N+1:end,:) = [];


function P = PermeanteLaturiDomeniuDiscretizat(geom,grid)
% domain permeance

% domain position
x1 = geom.x0;
x2 = geom.x0 + geom.lx;
z1 = geom.z0;
z2 = geom.z0 + geom.lz;

% domain discretization
nx = grid.nx;
nz = grid.nz;

vector_x = linspace(x1,x2,nx);
vector_z = linspace(z1,z2,nz);

N = nx*nz;
L = 2*N;     % number of edges
```

```
doinz = 2*nz;
P = zeros(L,1);

% browse the cells and add contributions to P_lat
for i = 1:nx-1
    r1 = vector_x(i);
    r2 = vector_x(i+1);
    for j = 1:nz-1
        k = (i-1)*nz + j;                    % node/cell index
        dz = vector_z(j+1) - vector_z(j);    % cell position celulei
        %[lambda_long_r1,lambda_long_r2,lambda_rad ] = PermeantaCelula(r1,r2,dz,geom.sim);
        lambda = PermeantaCelula(r1,r2,dz,geom.sim);
        lat_stg = 2*k;
        lat_jos = lat_stg - 1;
        lat_sus = lat_stg + 1;
        lat_dr = lat_stg + doinz;
        P(lat_stg) = P(lat_stg) + lambda.xmin;
        P(lat_dr) = P(lat_dr) + lambda.xmax;
        P(lat_jos) = P(lat_jos) + lambda.z;
        P(lat_sus) = P(lat_sus) + lambda.z;
    end
end
end


function lambda = PermeantaCelula(x1,x2,dz,sim)
% geometrical permeances of one cell
% between r1 and r2>r1>=0 and height dz>0

switch sim
    case 'axi'
        dr = x2 - x1;
        dr2 = dr/2;
        lambda.xmin = pi*((x1+dr2)^2 - x1^2)/dz;
        lambda.xmax = pi*(x2^2 - (x2-dr2)^2)/dz;
        if x1 < eps
%             x1 = eps;
%             lambda.z = pi*dz/(log(x2)-log(x1));
            lambda.z = 0;
        else
            lambda.z = pi*dz/(log(x2)-log(x1));
        end
    case 'xy'
        dx = x2 - x1;
        lambda.z = dz/(2*dx);
        lambda.xmin = dx/(2*dz);
        lambda.xmax = lambda.xmin;
    otherwise
        error('unknown symmetry, it should be either xy or axi');
end
end


function [out,Vterm1,Vterm2] = solve_EQS_axon_MsiC_omogene_izotrope(model)

% cytoplasm
modelC.grid.nx = model.grid.nx1;
modelC.grid.nz = model.grid.nz;
modelC.geom.x0 = model.geom.x0;
modelC.geom.z0 = model.geom.z0;
modelC.geom.lx = model.geom.a;
modelC.geom.lz = model.geom.l;
modelC.geom.sim = 'axi';
AC = MatriceIncidentaLaturiNoduri(modelC.grid);
PC = PermeanteLaturiDomeniuDiscretizat(modelC.geom,modelC.grid); % vector
LC = length(PC);
PC = spdiags(PC,0,LC,LC);
PnodalC = AC*PC*AC';
NC = length(PnodalC);

% membrane
modelM.grid.nx = model.grid.nx2;
modelM.grid.nz = model.grid.nz;
modelM.geom.x0 = model.geom.x0 + model.geom.a;
modelM.geom.z0 = model.geom.z0;
modelM.geom.lx = model.geom.b - model.geom.a;
modelM.geom.lz = model.geom.l;
modelM.geom.sim = 'axi';
AM = MatriceIncidentaLaturiNoduri(modelM.grid);
PM = PermeanteLaturiDomeniuDiscretizat(modelM.geom,modelM.grid); % vector
LM = length(PM);
PM = spdiags(PM,0,LM,LM);
PnodalM = AM*PM*AM';
NM = length(PnodalM);

%%% in the global numbering, we deduct the indices corresponding to the
%%% terminals, ground and shared nodes

idx_com = NC-model.grid.nz+1:NC;     % indices of the shared nodes
N = NC + NM - model.grid.nz;         % total number of nodes

idx_gnd = zeros(N,1);
i_gnd = 0;
idx_term = zeros(N,1);
```

```
i_term = 0;
not = length(model.term);
no_term_exc = 0;
excitationType = '';
for k = 1:not
    exc = model.term{k}.exc;
    switch exc
        case 'ECEv'
            if isempty(excitationType)
                excitationType = 'ECEv';
            elseif ~strcmp(excitationType,'ECEv')
                error('All terminals have to be excited in a similar way');
            end
            no_term_exc = no_term_exc + 1;
            indici = model.term{k}.idx;
            noi = length(indici);
            idx_term(i_term+1:i_term+noi) = indici;
            i_term = i_term+noi;
            if no_term_exc == 1
                lastidx = length(indici);
            end
        case 'ECEc'
            if isempty(excitationType)
                excitationType = 'ECEc';
            elseif ~strcmp(excitationType,'ECEc')
                error('All terminals have to be excited in a similar way');
            end
            no_term_exc = no_term_exc + 1;
            indici = model.term{k}.idx;
            noi = length(indici);
            idx_term(i_term+1:i_term+noi) = indici;
            i_term = i_term+noi;
            if no_term_exc == 1
                lastidx = length(indici);
            end
        case 'nonECEc'
            if isempty(excitationType)
                excitationType = 'nonECEc';
            elseif ~strcmp(excitationType,'nonECEc')
                error('All terminals have to be excited in a similar way');
            end
            no_term_exc = no_term_exc + 1;
            indici = model.term{k}.idx;
            noi = length(indici);
            idx_term(i_term+1:i_term+noi) = indici;
            i_term = i_term+noi;
            if no_term_exc == 1
                lastidx = length(indici);
            end
        case 'g'
            indici = model.term{k}.idx;
            noi = length(indici);
            idx_gnd(i_gnd+1:i_gnd+noi) = indici;
            i_gnd = i_gnd+noi;
        otherwise
            error('terminal type not implemented');
    end
end
idx_term(i_term+1:end) = [];
if no_term_exc ~= 2
    error('exactly two terminals have to be excited, in the same way: ECEv, ECEc, or nonECEc ');
end
no_term1 = lastidx;                     % number of nodes that belong to terminal 1
no_term2 = i_term - no_term1;           % number of nodes that belong to terminal 2
idx_term1 = idx_term(1:lastidx);        % indices of the nodes that belong to terminal 1
idx_term2 = idx_term(lastidx+1:end);    % indices of the nodes that belong to terminal 1

idx_gnd(i_gnd+1:end) = [];
idx_nec = 1:N;
idx_nec = idx_nec';
idx_nec([idx_term;idx_gnd]) = [];
no_nec = length(idx_nec);               % number of degrees of freedom
fprintf('No of DOFs = %d \n',no_nec);

%% assembling Gnodal mixed
Gnodal = sparse(N,N);
dimC = length(PnodalC);
Gnodal(1:dimC,1:dimC) = PnodalC*model.matC.sig;
dimcom = length(idx_com);               % number of shared nodes
idx = dimC-dimcom+1;
Gnodal(idx:end,idx:end) = Gnodal(idx:end,idx:end) + PnodalM*model.matM.sigR;

Gnn = Gnodal(idx_nec,idx_nec);
Gnt1 = Gnodal(idx_nec,idx_term1);
Gnt2 = Gnodal(idx_nec,idx_term2);
Gtt1 = Gnodal(idx_term1,idx_term1);
Gtt2 = Gnodal(idx_term2,idx_term2);
Gtt12 = Gnodal(idx_term1,idx_term2);

%% assembling Cnodal mixed
Cnodal = sparse(N,N);
Cnodal(idx:end,idx:end) = PnodalM*model.matM.eps;

Cnn = Cnodal(idx_nec,idx_nec);
```

```
Cnt1 = Cnodal(idx_nec,idx_term1);
Cnt2 = Cnodal(idx_nec,idx_term2);
Ctt1 = Cnodal(idx_term1,idx_term1);
Ctt2 = Cnodal(idx_term2,idx_term2);
Ctt12 = Cnodal(idx_term1,idx_term2);

nop = model.frecv.nop;
% frecvHz = linspace(model.frecv.fmin,model.frecv.fmax,nop);
frecvHz = logspace(log10(model.frecv.fmin),log10(model.frecv.fmax),nop);
frecvRad = frecvHz*2*pi;
frecv_resp = zeros(2,2,nop);

switch excitationType
    case 'ECEv'
        Vterm1 = [ones(no_term1,1) zeros(no_term1,1)]; % excitations of the two simulations
        Vterm2 = [zeros(no_term2,1) ones(no_term2,1)];

        disp('Excitation ECE in voltage, computing the matrix of transfer conductances');
        if no_nec == 0
            for k = 1:nop
                Ytt1  = Gtt1  + 1i*frecvRad(k)*Ctt1;
                Ytt12 = Gtt12 + 1i*frecvRad(k)*Ctt12;
                Ytt2  = Gtt2  + 1i*frecvRad(k)*Ctt2;

                It1 = Ytt1*Vterm1 + Ytt12*Vterm2;
                It2 = Ytt2*Vterm2 + conj(Ytt12)'*Vterm1;
                %Vnec = [];
                out = [sum(It1); sum(It2)]; % merging, the output represents complex admittances
                frecv_resp(:,:,k) = out;
            end
        else
            for k = 1:nop
                Ytt1 = Gtt1 + 1i*frecvRad(k)*Ctt1;
                Ytt12 = Gtt12 + 1i*frecvRad(k)*Ctt12;
                Ytt2 = Gtt2 + 1i*frecvRad(k)*Ctt2;
                Ynn = Gnn + 1i*frecvRad(k)*Cnn;
                Ynt1 = Gnt1 + 1i*frecvRad(k)*Cnt1;
                Ynt2 = Gnt2 + 1i*frecvRad(k)*Cnt2;

                Vnec = -Ynn\(Ynt1*Vterm1+Ynt2*Vterm2);
                It1 = Ytt1*Vterm1 + Ytt12*Vterm2 + conj(Ynt1)'*Vnec;
                It2 = Ytt2*Vterm2 + conj(Ytt12)'*Vterm1 + conj(Ynt2)'*Vnec;
                out = [sum(It1); sum(It2)]; % merging, the output represents complex admittances
                frecv_resp(:,:,k) = out;
            end
        end

    case 'ECEc'
        Gtt1 = sum(sum(Gtt1));  % merge before solving (constraint of equal potentials for the
            terminals nodes)
        Gtt2 = sum(sum(Gtt2));
        Gnt1 = sum(Gnt1,2);
        Gnt2 = sum(Gnt2,2);
        Gtt12 = sum(sum(Gtt12));
        Gnodal = [Gtt1 Gtt12 Gnt1';...
            Gtt12' Gtt2 Gnt2';...
            Gnt1   Gnt2 Gnn];
        %
        Ctt1 = sum(sum(Ctt1));  % merge before solving (constraint of equal potentials for the
            terminals nodes)
        Ctt2 = sum(sum(Ctt2));
        Cnt1 = sum(Cnt1,2);
        Cnt2 = sum(Cnt2,2);
        Ctt12 = sum(sum(Ctt12));
        Cnodal = [Ctt1 Ctt12 Cnt1';...
            Ctt12' Ctt2 Cnt2';...
            Cnt1   Cnt2 Cnn];
        %
        It1 = [1 0];
        It2 = [0 1];

        I = [It1; It2; zeros(no_nec,2)];
        disp('Excitation ECE in current, computing the matrix of transfer conductances');

        for k = 1:nop
            Ynodal = Gnodal + 1i*frecvRad(k)*Cnodal;
            x = Ynodal\I;
            Vterm1 = x(1,:);
            Vterm2 = x(2,:);
            %Vnec = x(3:end,:);
            out = [Vterm1; Vterm2];
            frecv_resp(:,:,k) = out;
        end

    case 'nonECEc'
        It1 = [ones(no_term1,1)/no_term1 zeros(no_term1,1)]; % excitations of the two simulations
        It2 = [zeros(no_term2,1) ones(no_term2,1)/no_term2];
        Gnodal = [Gtt1 Gtt12 Gnt1';...
            Gtt12' Gtt2 Gnt2';...
            Gnt1   Gnt2 Gnn];
        Cnodal = [Ctt1 Ctt12 Cnt1';...
            Ctt12' Ctt2 Cnt2';...
            Cnt1   Cnt2 Cnn];
        I = [It1; It2; zeros(no_nec,2)];
```

```
        % average voltage
        disp('Excitation nonECE in current, computing the matrix of transfer conductances, by averaging
              the potentials');
        for k = 1:nop
            Ynodal = Gnodal + 1i*frecvRad(k)*Cnodal;
            x = Ynodal\I;
            Vterm1 = x(1:no_term1,:);
            Vterm2 = x(no_term1+1:no_term1+no_term2,:);
            %Vnec = x(no_term1+no_term2+1:end,:);
            out = [sum(Vterm1)/no_term1; sum(Vterm2)/no_term2];
            frecv_resp(:,:,k) = out;
        end
    otherwise
        error('Unknown excitation type');
end
fprintf('        %d frequencies\n',nop);
filenameY_RI = strcat(model.frecv.filename,'_',excitationType,'_Y_RI.s2p');
filenameY_MA = strcat(model.frecv.filename,'_',excitationType,'_Y_MA.s2p');
filenameZ_RI = strcat(model.frecv.filename,'_',excitationType,'_Z_RI.s2p');
filenameZ_MA = strcat(model.frecv.filename,'_',excitationType,'_Z_MA.s2p');
filenameS_DB = strcat(model.frecv.filename,'_',excitationType,'_S_DB.s2p');
chdir(model.frecv.folder);
if strcmp(excitationType,'ECEv')
    writesnp_v2(filenameY_RI, frecvHz, frecv_resp, 'Y', 'Hz',50,'RI');
    snp2snp('',filenameY_RI,'Z','RI',filenameZ_RI);
else
    writesnp_v2(filenameZ_RI, frecvHz, frecv_resp, 'Z', 'Hz',50,'RI');
    snp2snp('',filenameZ_RI,'Y','RI',filenameY_RI);
end
snp2snp('',filenameY_RI,'Y','MA',filenameY_MA);
snp2snp('',filenameZ_RI,'Z','MA',filenameZ_MA);
snp2snp('',filenameY_RI,'S','DB',filenameS_DB);
chdir('..');
end
```

Functions for drawing

```
function draw_domain_axon(geom)
    xs = geom.x0;
    xd1 = geom.x0 + geom.a;
    xd2 = geom.x0 + geom.b;
    zs = geom.z0;
    zd = geom.z0 + geom.l;
    x = [xd1 xd1 xs xs xd1 xd2 xd2 xd1]*1e6;
    z = [zs zd zd zs zs zs zd zd]*1e6;
    plot(x,z,'-k','Linewidth',2);
    ylim([0.9*min(z),1.1*max(z)]);
    xlim([0,1.1*max(x)]);
    xlabel('r [{\mu}m]');
    ylabel('z [{\mu}m]');
    title('2.5 D, Oz is the axis or rotation');
    % hold on;
    % plot([0 0],[0.9*min(z),1.1*max(z)],'-r','Linewidth',1);
end


function draw_grid_axon(geom,grid)
    xs = geom.x0;
    xd1 = geom.x0 + geom.a;
    xd2 = geom.x0 + geom.b;
    zs = geom.z0;
    zd = geom.z0 + geom.l;
    nx1 = grid.nx1;
    nx2 = grid.nx2;
    nz = grid.nz;

    grid_x1 = linspace(xs,xd1,nx1);
    grid_x2 = linspace(xd1,xd2,nx2);
    grid_z = linspace(zs,zd,nz);
    grid_x = [grid_x1 grid_x2(2:end)];

    nx = nx1 + nx2 - 1;
    for i = 1:nx
        plot([grid_x(i) grid_x(i)]*1e6,[zs,zd]*1e6,':m');
    end
    for i = 1:nz
        plot([xs xd2]*1e6,[grid_z(i) grid_z(i)]*1e6,':m');
    end
end


function draw_terminals_axon(geom,term)
xs = geom.x0;
%xd1 = geom.x0 + geom.a;
xd2 = geom.x0 + geom.b;
zs = geom.z0;
zd = geom.z0 + geom.l;

not  = length(term); % no of terminals
for i = 1:not
    exc = term{i}.exc; % type of excitation (ECEv, ECEc, nonECEc,g)
    x = [term{i}.min term{i}.max];
```

```
    y = x;
    switch term{i}.poz
        case 'xmin'
            x = [xs xs];
        case 'xmax'
            x = [xd2 xd2];
        case 'zmin'
            y = [zs zs];
        case 'zmax'
            y = [zd zd];
        otherwise
            error('unknown pozition of terminal');
    end
    x = x*1e6;
    y = y*1e6;
    switch exc
        case 'ECEv'
            plot(x,y,'-r','Linewidth',3');
        case 'g'
            plot(x,y,'-k','Linewidth',6');
        case 'ECEc'
            plot(x,y,'-g','Linewidth',3');
        case 'nonECEc'
            plot(x,y,'-b','Linewidth',3');
        otherwise
            error('Unknown excitation of terminal');
    end
end
end


function number_nodes_axon(geom,grid)
    xs = geom.x0;
    xd1 = geom.x0 + geom.a;
    xd2 = geom.x0 + geom.b;
    zs = geom.z0;
    zd = geom.z0 + geom.l;
    nx1 = grid.nx1;
    nx2 = grid.nx2;
    nz = grid.nz;

    grid_x1 = linspace(xs,xd1,nx1);
    grid_x2 = linspace(xd1,xd2,nx2);
    grid_z = linspace(zs,zd,nz);
    grid_x = [grid_x1 grid_x2(2:end)];

    nx = nx1 + nx2 - 1;
    for i = 1:nx
        for j = 1:nz
            k = (i-1)*nz + j;
            text(grid_x(i)*1e6,grid_z(j)*1e6,num2str(k));
        end
    end
end


function show_nodes_terminals_axon(model)
    geom = model.geom;
    grid = model.grid;
    term = model.term;

    xs = geom.x0;
    xd1 = geom.x0 + geom.a;
    xd2 = geom.x0 + geom.b;
    zs = geom.z0;
    zd = geom.z0 + geom.l;
    nx1 = grid.nx1;
    nx2 = grid.nx2;
    nz = grid.nz;

    grid_x1 = linspace(xs,xd1,nx1);
    grid_x2 = linspace(xd1,xd2,nx2);
    grid_z = linspace(zs,zd,nz);
    grid_x = [grid_x1 grid_x2(2:end)];

    nx = nx1 + nx2 - 1;

    not  = length(term); % no of terminals
    for idx_t = 1:not
        exc = term{idx_t}.exc; % type of excitation (ECEv, ECEc, nonECEc, g)
        switch exc
            case 'ECEv'
                color = 'r';
            case 'g'
                color = 'm';
            case 'ECEc'
                color = 'g';
            case 'nonECEc'
                color = 'b';
            otherwise
                error('terminal type not implemented');
        end

        for k = model.term{idx_t}.idx
```

```
            j = rem(k,nz);
            if j == 0
                j = nz;
            end
            i = (k-j)/nz + 1;
            text(grid_x(i)*1e6,grid_z(j)*1e6,num2str(k),'Color',color);
        end
    end
end
```

snp functions

```
function [ok] = writesnp_v2(filename, freq_points, data, ptype, units,Z0,pformat)

% Saves to a snp file

% ------------------------------------------------------
% Writing header
% ------------------------------------------------------

[n,m,nm] = size(data);
if strcmpi(units,'HZ')
    freq_points = freq_points./1e9;
elseif strcmpi(units,'KHZ')
    freq_points = freq_points./1e6;
elseif strcmpi(units,'MHZ')
    freq_points = freq_points./1e3;
elseif strcmpi(units,'GHZ')
    %disp('Freq is in GHz');
else%  if strcmpi(units,'RAD/S')
    freq_points = freq_points./2/pi/1e9;
end;

units = 'GHz';

if m > 6
    nports = m;
    newname = strcat(filename,'.mat');
    pformat = 'RI';
    save(newname,'ptype','pformat','freq_points','data','Z0','units','nports','-mat');
    fprintf('!!!!!!!!! READ THIS !!!!!!!!! Output written in %s \n',newname);
    ok = 0;
else
    fid = fopen(filename,'w');
    if fid == -1
        error('Error in opening file %s',filename);
    end
    fprintf(fid,'%s %s %s %s %s %s','#',units,ptype,pformat,'R',num2str(Z0));

    dim = length(freq_points);
    ok=0;
    img = sqrt(-1); % just to be sure

    if m == 1
        A(:,1) = freq_points;
        data = squeeze(data);
        realpart = real(data);
        imagpart = imag(data);
        switch upper(pformat)
            case 'RI'
                A(:,2) = realpart;
                A(:,3) = imagpart;
            case 'MA'
                datac = realpart+img*imagpart;
                A(:,2) = abs(datac);
                A(:,3) = angle(datac)*180/pi;
            case 'DB'
                datac = realpart+img*imagpart;
                A(:,2) = 20*log10(abs(datac));
                A(:,3) = angle(datac)*180/pi;
        end
        for index = 1 : dim
            fprintf(fid, '\n%1.16e %1.16e %1.16e',A(index,1),A(index,2),A(index,3));
        end;
        %fprintf(fid, '\n%s','! end of data');
        fclose(fid);
        fprintf('Output written in %s \n',filename);
        fprintf('Current directory is %s \n',pwd);
    else % MIMO
        noval = m*m*2;
        A = zeros(nm,noval+1);
        A(:,1) = freq_points;
        index = 1;
        for j = 1:m
            for i = 1:m
                realpart = squeeze(real(data(i,j,:)));
                imagpart = squeeze(imag(data(i,j,:)));
                switch pformat
                    case 'RI'
                        index = index+1;
                        A(:,index) = realpart;
                        index = index+1;
```

```
                    A(:,index) = imagpart;
                case 'MA'
                    datac = realpart+img*imagpart;
                    index = index+1;
                    A(:,index) = abs(datac);
                    index = index+1;
                    A(:,index) = angle(datac)*180/pi;
                case 'DB'
                    datac = realpart+img*imagpart;
                    index = index+1;
                    A(:,index) = 20*log10(abs(datac));
                    index = index + 1;
                    A(:,index) = angle(datac)*180/pi;
            end
        end
    end

    for k = 1 : dim
        fprintf(fid, '\n');
        fprintf(fid, '%1.16e ',A(k,1));
        %fprintf(fid, '%2.20g ',A(k,1));
        for i = 2:noval+1
            fprintf(fid, ' %1.16e ',A(k,i));
            %fprintf(fid, ' %2.20g ',A(k,i));
        end
    end;
    %fprintf(fid, '\n%s','! end of data');
    fclose(fid);
    fprintf('Output written in %s \n',filename);
    fprintf('       in the folder %s \n',pwd);
end;
end


function [] = snp2snp(snp_pathname,snp_filename,type,representation,...
    new_filename)

numein = snp_filename;
[ptype,pformatfile,frequency_points,value,Z0,...
    freq_unit,nports] = ...
    loadsnp(strcat(snp_pathname,snp_filename));
ext = strcat('.s',num2str(nports),'p');
lext = length(ext);

if isempty(new_filename)
    new_filename = numein;
    ll = length(new_filename);
    newf = new_filename;
    new_filename = strcat(newf(1:ll-length(ext)),'_from_',ptype,...
        '_to_',type,ext);
else
    ll = length(new_filename);
    if ll > lext
        if ~strcmp(new_filename(ll-lext+1:ll),ext)
            % wrong extension - add correct extension
            new_filename = strcat(new_filename,ext);
        end
    else
        new_filename = strcat(new_filename,ext);
    end
end

if strcmp(ptype,type) ==0
    disp(sprintf('Conversion from %s to %s',ptype,type));
    switch ptype
        case 'S'
            switch type
                case 'Z'
                    [value] = mys2z(value,Z0);
                case 'Y'
                    [value] = mys2z(value,Z0);
                    [value] = myz2y(value);
                otherwise
                    disp('snp2snp.m: Invalid parameter. Only S, Z or Y implemented');
            end
        case 'Z'
            switch type
                case 'S'
                    [value] = myz2s(value,Z0);
                case 'Y'
                    [value] = myz2y(value);
                otherwise
                    disp('snp2snp.m: Invalid parameter. Only S, Z or Y implemented');
            end
        case 'Y'
            switch type
                case 'S'
                    [value] = myy2z(value);
                    [value] = myz2s(value,Z0);
                case 'Z'
                    [value] = myy2z(value);
                otherwise
                    disp('snp2snp.m: Invalid parameter. Only S, Z or Y implemented');
            end
```

```
        otherwise
            disp(sprintf('Format %s in file %s',ptype,snp_pathname));
            disp('Compute inverse');
            [value] = myy2z(value);
            type = 'H';
            new_filename = strcat(newf(1:ll-length(ext)),'_from_',ptype,...
                '_to_',type,ext);
    end
end

[ok] = writesnp_v2(strcat(snp_pathname,new_filename), frequency_points, value, type, freq_unit,Z0,
    representation);
if ok == 0
    disp(sprintf('%s file was written succesfully',new_filename));
    disp('snp2snp - DONE');
else
    disp('snp2snp - FAILURE');
end
```

A2. 2.5D – EC analytical solution, nonhomogeneous case

```
clear all;
close all hidden;
clc;
date.a = 7e-6 ;              % electrode radius
date.b = 10e-6;              % axons outer radius
date.L = 223e-6;             % length of axon compartment
date.n = 500 ;               % number of harmonics
date.m = 20;                 % number of radial samples
date.sig1 = 1.0824;          % cytoplasm conductivity
date.sig2 = 1.04e-4;         % membrane conductivity
eps0 = 8.854187817e-12;      % F/m
date.eps2 = 15.44*eps0 ;     % membrane absolute permittivity
%
% parameters used by the nonlinear procedure to compute lambda
date.maxit = 200;
date.errel = 1e-15;
%
date.f = 0; % stationary
%err = verifica2(date);      % with common factor D
err = verifica2_v2(date);    % with common factor C - also works in homogeneous case

format long e;
[Z11r0,Z21r0,Z11m,Z21m] = impedantaNeomogena(date)


function err = verifica2_v2(date)
    a = date.a;
    b = date.b;
    a2 = a^2;
    b2 = b^2;
    % L = date.L;
    n = date.n;
    m = date.m;
    sig1 = date.sig1;
    sig2 = date.sig2;
    eps2 = date.eps2;
    f = date.f;
    omega = 2*pi*f;
    beta = (sig2 + 1i*omega*eps2)/sig1;

    r = linspace(0,b,m) ;            % array of radial samples
    lk = valproprii(n,date,'yes');  % lambda_k
    lkb = lk*b;
    lka = lk*a;

    J0a = besselj(0,lka);
    J1a = besselj(1,lka);
    J0b = besselj(0,lkb);
    J1b = besselj(1,lkb);
    %
    Y0a = bessely(0,lka);
    Y1a = bessely(1,lka);
    Y0b = bessely(0,lkb);
    Y1b = bessely(1,lkb);

    DpeC_v1 = - J0b./Y0b;
    DpeC_v2 = (1-beta)*J0a.*J1a./(beta*J0a.*Y1a - J1a.*Y0a);
    DpeC = (DpeC_v1+DpeC_v2)/2;

    BpeC_v1 = 1 + DpeC.* Y0a./J0a;
    BpeC_v2 = beta*(1 + DpeC.*Y1a./J1a);
    BpeC = (BpeC_v1+BpeC_v2)/2;

    figure(101); clf;
    plot(abs(DpeC_v1-DpeC_v2),'-m');
    figure(102); clf;
    plot(abs(DpeC_v1-DpeC_v2)./abs(DpeC),'--m');

    figure(101); hold on;
```

```
    plot(abs(BpeC_v1-BpeC_v2),'-k');
    legend('Dif abs in DpC','Dif abs in BpC');
    figure(102); hold on;
    plot(abs(BpeC_v1-BpeC_v2)./abs(BpeC),'--k');
    legend('Dif rel in DpC','Dif rel in BpC');

    J0a2 = J0a.^2;
    J1a2 = J1a.^2;
    J0b2 = J0b.^2;
    J1b2 = J1b.^2;
    Y0a2 = Y0a.^2;
    Y1a2 = Y1a.^2;
    Y0b2 = Y0b.^2;
    Y1b2 = Y1b.^2;

    term1 = a2/2.*(J0a2 + J1a2);
    %
    term2JJ = (b2*(J0b2 + J1b2) - a2*(J0a2 + J1a2))/2;
    term2YY = (b2*(Y0b2 + Y1b2) - a2*(Y0a2 + Y1a2))/2;
    term2JY = (b2*(J0b.*Y0b + J1b.*Y1b) - a2*(J0a.*Y0a + J1a.*Y1a))/2;
    term2 = term2JJ + DpeC.^2.*term2YY + 2*DpeC.*term2JY;
    %
    Rk2 = sig1*BpeC.^2.*term1 + sig2*term2;

    F = BpeC.*J1a./(lk.*Rk2);
    F = F/(pi*a);
    ft = zeros(1,m);
    h = zeros(1,m);   % exact function (unitary pulse with support 0<r<a)
    for j = 1:m
        lr = lk*r(j);
        if r(j) < a
            h(j) = 1;
            R = BpeC.*besselj(0,lr);
        else
            R = besselj(0,lr) + DpeC.*bessely(0,lr);
        end
        ft(j) = sum(F.*R);
    end
    ft = ft*pi*a2*sig1;
    err = sum((h-ft).^2); % error
    err = sqrt(err/m);
    figure(10); clf;
    plot(r*1e6,h,'--b'); hold on;
    plot(r*1e6,ft,'-*r');
    xlabel('r [{\mu}m]');
    legend('h','ft');
    %ylim([0,1.2]);
    title(sprintf('Verification of the truncated Fourier series for the function 1(a-r) where a = %g um
        ',a*1e6));
end


function [Z11r0,Z21r0,Z11m,Z21m] = impedantaNeomogena(date)
    a = date.a;
    b = date.b;
    a2 = a^2;
    b2 = b^2;
    L = date.L;
    n = date.n;
    m = date.m;
    sig1 = date.sig1;
    sig2 = date.sig2;
    eps2 = date.eps2;
    f = date.f;
    omega = 2*pi*f;
    beta = (sig2 + 1i*omega*eps2)/sig1;

    r = linspace(0,a,m) ; % array of radial samples on the terminal, to compute the average
    lk = valproprii(n,date,'yes'); % lambda_k
    lkb = lk*b;
    lka = lk*a;
    %
    J0a = besselj(0,lka);
    J1a = besselj(1,lka);
    J0b = besselj(0,lkb);
    J1b = besselj(1,lkb);
    %
    Y0a = bessely(0,lka);
    Y1a = bessely(1,lka);
    Y0b = bessely(0,lkb);
    Y1b = bessely(1,lkb);
    %
    DpeC_v1 = - J0b./Y0b;
    DpeC_v2 = (1-beta)*J0a.*J1a./(beta*J0a.*Y1a - J1a.*Y0a);
    DpeC = (DpeC_v1+DpeC_v2)/2;
    %
    BpeC_v1 = 1 + DpeC.* Y0a./J0a;
    BpeC_v2 = beta*(1 + DpeC.*Y1a./J1a);
    BpeC = (BpeC_v1+BpeC_v2)/2;
    %
    J0a2 = J0a.^2;
    J1a2 = J1a.^2;
    J0b2 = J0b.^2;
    J1b2 = J1b.^2;
```

```
    Y0a2 = Y0a.^2;
    Y1a2 = Y1a.^2;
    Y0b2 = Y0b.^2;
    Y1b2 = Y1b.^2;
    %
    term1 = a2/2.*(J0a2 + J1a2);
    %
    term2JJ = (b2*(J0b2 + J1b2) - a2*(J0a2 + J1a2))/2;
    term2YY = (b2*(Y0b2 + Y1b2) - a2*(Y0a2 + Y1a2))/2;
    term2JY = (b2*(J0b.*Y0b + J1b.*Y1b) - a2*(J0a.*Y0a + J1a.*Y1a))/2;
    term2 = term2JJ + DpeC.^2.*term2YY + 2*DpeC.*term2JY;
    %
    Rk2 = sig1*BpeC.^2.*term1 + sig2*term2;
    %
    F = BpeC.*J1a./(lk.*Rk2);
    F = F/(pi*a);
    %
    lkL = lk*L;
    Cx0 = F./(lk.*tanh(lkL));
    CxL = F./(lk.*sinh(lkL));

    ftx0 = zeros(1,m);
    ftxL = zeros(1,m);
    for j = 1:m
        lr = lk*r(j);
        if r(j) < a
            R = BpeC.*besselj(0,lr);
        else
            R = besselj(0,lr) + DpeC.*bessely(0,lr);
        end
        ftx0(j) = sum(Cx0.*R);
        ftxL(j) = sum(CxL.*R);
    end
    figure(11); clf;
    plot(r*1e6,ftx0,'--b');
    title('Potential at x=0');

    figure(12); clf;
    plot(r*1e6,ftxL,'-*r');
    xlabel('r [{\mu}m]');
    title('Potential at x=L');

    % values for r = 0
    Z11r0 = ftx0(1);
    Z21r0 = ftxL(1);

    % averaged values
    grid_x1 = r'; % points of discretization for the terminal
    grid_x1dupl = [grid_x1 grid_x1];
    V1oriR = [ftx0' ftxL'].*grid_x1dupl;
    no_term1 = m;
    apatrat = a^2;
    coef = 2/apatrat*(grid_x1(end)-grid_x1(end-1));
    Vmediu11 = coef*(sum(V1oriR(2:no_term1-1,1))+ V1oriR(end,1)/2 + V1oriR(1,1)/2); % trapezes
    Vmediu21 = coef*(sum(V1oriR(2:no_term1-1,2))+ V1oriR(end,2)/2 + V1oriR(1,2)/2);
    Z11m = Vmediu11;
    Z21m = Vmediu21;
end
```

A3. Saltatory conduction – Coupled model generation in Matlab (Spice circuit of interconnected blocks)

```
function [] = main_createcir_VF_scaled_forSpice(N)

% main_createcir_VF_scaled_forSpice = automatic generation of circuit netlist for
%             N*(NonLIN-LIN)-NonLIN_OPEN
% input = number of segments NonLIN-LIN
% output = Spice netlist
%   idxnodfinal - index of the final node,
%                   -1 if file creation failed;
%
%   NODE:        circuit HH (nonlinear)
%   INTERNODE:   circuit extracted from VF (linear, reduced)
%
% Name of file: main_createcir_VF_scaled_forSpice.m
% Version: 1.0
% Last update: 11 July 2018
%==========================
%
idxnodes = {};
filename = strcat('circuit',num2str(N),'segm_spice3.cir');

% set parameters
ArieComuna = 200e-6; % cm^2
params.Cm = 1;    % uF/cm^2
params.Arie_Cm = ArieComuna;     % cm^2
params.gbarl = 0.3;
```

```
params.Arie_gbarl = ArieComuna;     % cm^2
params.gbarK = 36;
params.Arie_gbarK = ArieComuna;     % cm^2
params.gbarNa = 120;
params.Arie_gbarNa = ArieComuna;     % cm^2

% Arie_I1 = 75e-6;   % cm^2

% create NODE subcircuit (HH model)
createcirNODE_scaled_forSpice(params);

fid=fopen(filename,'w');
fprintf(fid,...
    '*********************************************************************************\n');
fprintf(fid,...
    '* SPICE circuit of %i*(NonLIN-LIN)-NonLIN_OPEN\n',N);
fprintf(fid,...
    '* Number of segments NonLIN-LIN = %i\n',N);
fprintf(fid,...
    '*********************************************************************************\n');

fprintf(fid,'\n \n');

fprintf(fid,'.tran 0.03 30\n');
fprintf(fid,'I1 0 2 PULSE(0 2e-2 10 1e-9 1e-9 5 30 0)\n');
%fprintf(fid,'I1 0 2 PULSE(0 {10*Arie_I1} 10 1e-9 1e-9 5 30 30)\n');
fprintf(fid,'.inc subcktNODE_Comsol.cir\n');
fprintf(fid,'.inc internode_VF_L1_scaled.cir\n');

% fprintf(fid,'R2 1 0 1E100\n');

idx = 1;    % index root node for NonLIN-LIN segment, for the first segment is 1
idxs = 1;   % index root subcircuit for NonLIN-LIN segment, for the first segment is 1
for i=1:N
    fprintf(fid,'\n*node%i\n',i);
    fprintf(fid,'xsub%i %i %i %iout NODE\n',idxs,0,idx+1,idx+3);
    fprintf(fid,'Vaux%i1 %iout %iin -80\n',idxs,idx+3,idx+3);
    fprintf(fid,'xsub%i %iin %iout %i INTERNODE_VF_L1_scaled\n',idxs+1,idx+3,idx+5,0);
    fprintf(fid,'Vaux%i2 %iout %i 80\n',idxs,idx+5,idx+5);
    idxnodes{i} = strcat(num2str(idx+3),'out');
    idx = idx + 4;
    idxs = idxs + 2;
end

% last bloc: NonLIN open-circuit
fprintf(fid,'\n*node%i\n',N+1);
fprintf(fid,'xsub%i %i %i %i NODE\n',idxs,0,idx+1,idx+3);

fprintf(fid,'.END\n');
fprintf(fid,'\n');

fclose(fid);
idxnodes{N+1} = num2str(idx+3);

fprintf('\nCreation of SPICE file (circuit (%i*NonLIN-LIN)-NonLIN_OPEN) - DONE',N);
fprintf('\n\nIndices of output nodes for NonLIN are: \n');
disp(idxnodes);


function [ok] = createcirNODE_scaled_forSpice(params)

% createcirNODE_scaled_forSpice = automatic generation of circuit netlist as subcircuit for
%              an HH model (with parameters from Comsol model)
%
%           global model, scaled quantities
%
% output = Spice netlist
%   ok - output flag; ok = 0 means normal (successful) top
%
% Name of file: createcirNODE_scaled_forSpice.m
% Version: 1.0
% Last update: 11 July 2018
%==========================
%
ok = 0;
filename = 'subcktNODE_Comsol.cir';

fid=fopen(filename,'w');
fprintf(fid,...
    '*********************************************************************************\n');
fprintf(fid,...
    '* SPICE circuit of NODE\n');
fprintf(fid,...
    '* HH model with parameters from Comsol model\n');
fprintf(fid,...
    '*********************************************************************************\n');
fprintf(fid,'* time is in [ms]\n');
fprintf(fid,'* equations are scaled and global\n');
fprintf(fid,'* currents are in [uA]\n');
fprintf(fid,'* potentials are in [mV]\n');
fprintf(fid,'\n \n');

fprintf(fid,'.SUBCKT NODE 100 99 102\n');
```

```
fprintf(fid,'CE 99 100 %2.5e IC=-80\n',params.Cm*params.Arie_Cm);
fprintf(fid,'CN n 0 1 IC=0.317\n');
fprintf(fid,'B1_GAN 0 n I=0.01*((V(99)-V(100))+55)/(1-exp(-((V(99)-V(100))+55)/10))*(1-V(n))\n');
fprintf(fid,'B2_GBN 0 n I=-0.125*exp(-((V(99)-V(100))+65)/80)*V(n)\n');

fprintf(fid,'CM m 0 1 IC=0.052\n');
fprintf(fid,'B2_GAM 0 m I=0.1 *((V(99)-V(100))+40)/(1-exp(-((V(99)-V(100))+40)/10))*(1-V(m))\n');
fprintf(fid,'B1_GBM 0 m I=-4*exp(-0.0556*((V(99)-V(100))+65))*V(m)\n');

fprintf(fid,'CH h 0 1 IC=0.596\n');
fprintf(fid,'B1_GBH 0 h I=0.07*exp(-0.05*((V(99)-V(100))+65))*(1-V(h))\n');
fprintf(fid,'B2_GAH 0 h I=-1/(1+exp(-((V(99)-V(100))+35)/10))*V(h)\n');

fprintf(fid,'Rleak 99 Naux %2.5e\n',...
    1/(params.gbarl*params.Arie_gbarl));
fprintf(fid,'Vleak Naux 100 -54.4\n');

fprintf(fid,...
    'B1_GK 99 100 I=%2.5e*((V(99)-V(100))-(-77))*V(n)*V(n)*V(n)*V(n)\n',...
    params.gbarK*params.Arie_gbarK);
fprintf(fid,...
    'B2_GNa 99 100 I=%2.5e*((V(99)-V(100))-50)*V(m)*V(m)*V(m)*V(h)\n',...
    params.gbarNa*params.Arie_gbarNa);

fprintf(fid,'Vaux1 99 102 0\n');
%fprintf(fid,'Raux2 100 0 1e-22\n');

fprintf(fid,'.ENDS\n');
fprintf(fid,'\n');

fclose(fid);
ok = 1;

disp('Creation of SPICE file (subcircuit NODE) - DONE');
```

Reduction procedure (POD)

```
function reduce_v2()

% Name of file: reduce_v2.m
% Version: 2.0
% Last update: 23 July 2018
%=========================

nr_poles_max = 70;
filename.V = 'tabel55_500_320.txt';
% filename.n = 'n_de_t.txt';
% filename.m = 'm_de_t.txt';
% filename.h = 'h_de_t.txt';
dim.dim1 = 2049;     % 14 % 555 % 626 % 1648 % 3561 % timp
dim.dim2 = 501;      % number of state variables
dim.slice = 501;
Ferr_rel_F_Fredus_mijloc = zeros(nr_poles_max,1);
Ferr_rel_F_Fredus_end = zeros(nr_poles_max,1);
legm = {}; lege = {}; leg = {};
time = linspace(0,320,dim.dim1);
% change the format here
% fm = {'o-.','s--','p--','*--','d--','o--','s-.','p-.'};
fm = {'--','--','--','--','--','--','--','--','--','--'};
co = get(gca,'colororder');
tic
[A,svdm] = preprocesare(filename,dim);
toc

idx = 0;
for k=1:nr_poles_max
    tic
    [~,respFredus] = morPODbySVD(A,svdm,dim,k);
    toc
    [~, Ferr_rel_F_Fredus_mijloc(k)] = getErrors(A((dim.slice-1)/2,:),respFredus.mijloc);
    [~, Ferr_rel_F_Fredus_end(k)] = getErrors(A(dim.slice,:),respFredus.end);

    fprintf('Relative error between Fexact and Fredus (%i poli): (from time resp.) middle: %2.2e, final
        : %2.2e\n'...
        ,k,Ferr_rel_F_Fredus_mijloc(k), Ferr_rel_F_Fredus_end(k));

    if (mod(k,10)==0)
        idx = idx + 1;
        figure(100);
        plot(time,respFredus.mijloc,fm{idx},'LineWidth',2.5,...
            'Color',co(idx,:));
        legm{length(legm)+1} = ...
            strcat('Red. model (middle node)-order ',num2str(k));
        hold on;
%         figure(200);
        hh = plot(time,respFredus.end,fm{idx},'LineWidth',2.5,...
            'Color',co(idx,:));
        legv(idx) = hh;
        lege{length(lege)+1} = ...
            strcat('Red. model (middle & last node)-order ',num2str(k));
        hold on;
    end
end
```

```
figure(100);
plot(time,A((dim.slice-1)/2,:),'k','LineWidth',2);
% legm{length(legm)+1}=' Full model (middle & last node)';
% xlabel('time [ms]');
% ylabel('Electric potential [mV]');
% ll=legend(legm); ll.FontSize=14; set(gca,'FontSize',16);
% grid on;
hold on;

% figure(200);
legv(end+1) = plot(time,A(dim.slice,:),'k','LineWidth',2);
lege{length(lege)+1}=' Full model (middle & last node)';
xlabel('time [ms]');
ylabel('Electric potential [mV]');
ll=legend(legv',lege); ll.FontSize=14; set(gca,'FontSize',16);
grid on;

figure(300);
semilogy(...
    1:nr_poles_max,Ferr_rel_F_Fredus_mijloc,'o'...
    ,'MarkerSize', 8,'MarkerEdgeColor','k'...
    ,'MarkerFaceColor','b');
hold on
semilogy(...
    1:nr_poles_max,Ferr_rel_F_Fredus_end,'o'...
    ,'MarkerSize', 8,'MarkerEdgeColor','k'...
    ,'MarkerFaceColor','r');
grid on
leg{length(leg)+1} = 'Relative error between Fe and Fa - middle node';
leg{length(leg)+1} = 'Relative error between Fe and Fa - last node';
legl=legend(leg); legl.FontSize = 16; set(gca,'FontSize',16);
xlabel('Order of approximation (no. of poles)');
ylabel('Relative error [-]');

end


function [domain,respFPODbySVD] = morPODbySVD(A, svdm, dim, order)
    fprintf('\n Reduce model with POD by SVD ... \n');

    U = svdm.U;
    S = svdm.S;
    V = svdm.V;

    %% build matrix of small rank
    X = zeros(size(A));          % rank(X) <= r
    for i=1:order
        X = X + S(i,i)*kron(U(:,i),V(:,i)');
        fprintf('idx=%i  norm: %2.3e, current_singular_value: %2.3e \n'...
            ,i,norm(A-X), S(i,i));
    end

%     respFPODbySVD = X(end,:);
    respFPODbySVD.mijloc = X((dim.slice-1)/2,:);
    respFPODbySVD.end = X(dim.slice,:);

    %% find optimal rank(X)
    % build X so that sigma_(k+1)(nA) < ||A-X||_2 < sigma_(k)(A)
    % therefore rank(X) = k ;
    X = ones(size(A));          % rank(X) <= r
    X = X + S(1,1)*kron(U(:,1),V(:,1)');

    idx = 2;
    nritmax = size(A,1);
    while ( (norm(A-X) > S(idx-1,idx-1) ...
             || norm(A-X) < S(idx,idx)) ...
            && idx < nritmax )
        X = zeros(size(A));
        % build X using first idx singular values of A
        for i = 1:idx
            X = X + S(i,i)*kron(U(:,i),V(:,i)');
        end

        fprintf('idx=%i  norm: %2.3e, current_singular_value: %2.3e \n'...
            ,idx,norm(A-X), S(idx,idx));
        idx = idx+1;
    end

    fprintf('Optimal rank of X is %i \n', rank(X));

    domain = 'time';

    fprintf('\n Reduce model with POD by SVD: DONE \n');
end


function isMatrixUnitary(U,str)
    % verifies U is unitary
    Uinverse = inv(U);
    UConjTran = U';
    tol=10e-6;
    er=abs(UConjTran-Uinverse);
    if sum(er(:))<tol
```

```
        fprintf('\nMatrix %s is unitary\n',str)
    else
        fprintf('\nMatrix %s is NOT unitary\n',str)
    end
end


function [A,svdm] = preprocesare(filename,dim)
    % get time-domain response matrix
    A = readSamplesMatrix(filename,dim);
    fprintf('Size of matrix A is: %ix%i\n',size(A,1),size(A,2));

    % use svd to find singular values (sigma) and vectors of A
    [U,S,V]= svd(A);

    r = rank(S);
    fprintf('Initial rank of matrix A is: %i\n',r);

    %%% verifications
    Asvd = zeros(size(A));
    % rebuild A into Asvd - for verification purposes
    for i = 1:r
        Asvd = Asvd + S(i,i)*kron(U(:,i),V(:,i)');
    end

    norm_diff = norm(A-Asvd);
    fprintf('Verification: norm(A-Asvd) rebuild %2.3e \n',norm_diff);
    isMatrixUnitary(U,'U');
    isMatrixUnitary(V,'V');

    svdm.U = U;
    svdm.S = S;
    svdm.V = V;
end


function A = readSamplesMatrix(filename,dim)
    An = dlmread(filename.V,'    ');

    % modify here corresponding to the number of segments
%     A = zeros(dim.dim2,dim.dim1);
 %  idcs = 4:4:400; idcs(end+1) = 404;
    idcs = 4:4:2000; idcs(end+1) = 2002;
%    idcs = 4:4:948;
    A = An(1:dim.dim1,idcs)';

    % A = reshape(An(:,1),[dim.dim1,dim.dim1])';
end


function [err_abs,err_rel]=getErrors(v1,v2)
    % GETERRORS Absolute and relative error between two vectors
    % Compute the error based on Euclidean norm
    l=length(v1);

    % get index of max
    [maxim,indexmax] = max(v1);

    err_abs=sqrt(sum(abs(v1-v2).^ 2) / l );
    err_rel=abs(err_abs/v1(indexmax));
end
```

Generated circuit for Spice – N=13 sections

(INTERNODE circuit is previously extracted from VF)

```
********************************************************************************
* SPICE circuit of 13*(NonLIN-LIN)-NonLIN_OPEN
* Number of segments NonLIN-LIN = 13
********************************************************************************

.tran 0.03 30
I1 0 2 PULSE(0 2e-2 10 1e-9 1e-9 5 30 0)
.inc subcktNODE_Comsol.cir
.inc internode_VF_L1_scaled.cir

*node1
xsub1 0 2 4out NODE
Vaux11 4out 4in -80
xsub2 4in 6out 0 INTERNODE_VF_L1_scaled
Vaux12 6out 6 80

*node2
xsub3 0 6 8out NODE
Vaux31 8out 8in -80
xsub4 8in 10out 0 INTERNODE_VF_L1_scaled
Vaux32 10out 10 80

*node3
xsub5 0 10 12out NODE
Vaux51 12out 12in -80
```

```
xsub6 12in 14out 0 INTERNODE_VF_L1_scaled
Vaux52 14out 14 80

*node4
xsub7 0 14 16out NODE
Vaux71 16out 16in -80
xsub8 16in 18out 0 INTERNODE_VF_L1_scaled
Vaux72 18out 18 80

*node5
xsub9 0 18 20out NODE
Vaux91 20out 20in -80
xsub10 20in 22out 0 INTERNODE_VF_L1_scaled
Vaux92 22out 22 80

*node6
xsub11 0 22 24out NODE
Vaux111 24out 24in -80
xsub12 24in 26out 0 INTERNODE_VF_L1_scaled
Vaux112 26out 26 80

*node7
xsub13 0 26 28out NODE
Vaux131 28out 28in -80
xsub14 28in 30out 0 INTERNODE_VF_L1_scaled
Vaux132 30out 30 80

*node8
xsub15 0 30 32out NODE
Vaux151 32out 32in -80
xsub16 32in 34out 0 INTERNODE_VF_L1_scaled
Vaux152 34out 34 80

*node9
xsub17 0 34 36out NODE
Vaux171 36out 36in -80
xsub18 36in 38out 0 INTERNODE_VF_L1_scaled
Vaux172 38out 38 80

*node10
xsub19 0 38 40out NODE
Vaux191 40out 40in -80
xsub20 40in 42out 0 INTERNODE_VF_L1_scaled
Vaux192 42out 42 80

*node11
xsub21 0 42 44out NODE
Vaux211 44out 44in -80
xsub22 44in 46out 0 INTERNODE_VF_L1_scaled
Vaux212 46out 46 80

*node12
xsub23 0 46 48out NODE
Vaux231 48out 48in -80
xsub24 48in 50out 0 INTERNODE_VF_L1_scaled
Vaux232 50out 50 80

*node13
xsub25 0 50 52out NODE
Vaux251 52out 52in -80
xsub26 52in 54out 0 INTERNODE_VF_L1_scaled
Vaux252 54out 54 80

*node14
xsub27 0 54 56 NODE
.END


********************************************************************************
* SPICE circuit of NODE
* HH model with parameters from Comsol model
********************************************************************************
* time is in [ms]
* equations are scaled and global
* currents are in [uA]
* potentials are in [mV]

.SUBCKT NODE 100 99 102
CE 99 100 2.00000e-04 IC=-80
CN n 0 1 IC=0.317
B1_GAN 0 n I=0.01*((V(99)-V(100))+55)/(1-exp(-((V(99)-V(100))+55)/10))*(1-V(n))
B2_GBN 0 n I=-0.125*exp(-((V(99)-V(100))+65)/80)*V(n)
CM m 0 1 IC=0.052
B2_GAM 0 m I=0.1 *((V(99)-V(100))+40)/(1-exp(-((V(99)-V(100))+40)/10))*(1-V(m))
B1_GBM 0 m I=-4*exp(-0.0556*((V(99)-V(100))+65))*V(m)
CH h 0 1 IC=0.596
B1_GBH 0 h I=0.07*exp(-0.05*((V(99)-V(100))+65))*(1-V(h))
B2_GAH 0 h I=-1/(1+exp(-((V(99)-V(100))+35)/10))*V(h)
Rleak 99 Naux 1.66667e+04
Vleak Naux 100 -54.4
B1_GK 99 100 I=7.20000e-03*((V(99)-V(100))-(-77))*V(n)*V(n)*V(n)*V(n)
B2_GNa 99 100 I=2.40000e-02*((V(99)-V(100))-50)*V(m)*V(m)*V(m)*V(h)
Vaux1 99 102 0
.ENDS
```

```
********************************************************************************
* SPICE equivalent circuit generated by Codestar snp2cir developed in UPB/LMN
********************************************************************************
*Differential-equation macromodel

.SUBCKT INTERNODE_VF_L1_scaled Term1  Term2   OUT

Ginf1_1 Term1 OUT Term1 OUT 1.793323e-02
Greal1_1_1 Term1 OUT nod1_1 0 -4.916820e+03
Greal1_1_2 Term1 OUT nod1_2 0 -7.790886e+01
Greal1_1_3 Term1 OUT nod1_3 0 -2.189824e+01
Ginf1_2 Term1 OUT Term2 OUT 3.374289e-04
Greal1_2_1 Term1 OUT nod2_1 0 -2.137527e+02
Greal1_2_2 Term1 OUT nod2_2 0 5.359397e+01
Greal1_2_3 Term1 OUT nod2_3 0 -2.244383e+01
Ginf2_1 Term2 OUT Term1 OUT 3.374289e-04
Greal2_1_1 Term2 OUT nod1_1 0 -2.137527e+02
Greal2_1_2 Term2 OUT nod1_2 0 5.359397e+01
Greal2_1_3 Term2 OUT nod1_3 0 -2.244383e+01
Ginf2_2 Term2 OUT Term2 OUT 1.793323e-02
Greal2_2_1 Term2 OUT nod2_1 0 -4.916820e+03
Greal2_2_2 Term2 OUT nod2_2 0 -7.790886e+01
Greal2_2_3 Term2 OUT nod2_3 0 -2.189824e+01

*elements corresponding to the real poles:
*=========================================:

*Real pole circuit 1 1
Ga1_1 0 nod1_1 Term1 OUT 1
Ca1_1 nod1_1 0 1
Ra1_1 nod1_1 0 2.865263e-06

*Real pole circuit 1 2
Ga1_2 0 nod1_2 Term1 OUT 1
Ca1_2 nod1_2 0 1
Ra1_2 nod1_2 0 1.885642e-05

*Real pole circuit 1 3
Ga1_3 0 nod1_3 Term1 OUT 1
Ca1_3 nod1_3 0 1
Ra1_3 nod1_3 0 6.211166e-05

*Real pole circuit 2 1
Ga2_1 0 nod2_1 Term2 OUT 1
Ca2_1 nod2_1 0 1
Ra2_1 nod2_1 0 2.865263e-06

*Real pole circuit 2 2
Ga2_2 0 nod2_2 Term2 OUT 1
Ca2_2 nod2_2 0 1
Ra2_2 nod2_2 0 1.885642e-05

*Real pole circuit 2 3
Ga2_3 0 nod2_3 Term2 OUT 1
Ca2_3 nod2_3 0 1
Ra2_3 nod2_3 0 6.211166e-05

*elements corresponding to the complex poles:
*=========================================:

.ENDS INTERNODE_VF_L1_scaled

************************
*Example of Main circuit:
************************
*V1 Term1 0 AC 1
*V2 Term2 0 AC 0

*X1 Term1  Term2   0 SYNT

*.PROBE
*.AC LIN 100  6.283185e+00 6.283185e+04
*.END
```

References

[1] SIAM Working Group on CSE Education. Graduate education in computational science and engineering. *SIAM Review*, 43(1):163–177, 2001.

[2] Ulrich Rüde, Karen Willcox, Lois Curfman McInnes, and Hans De Sterck. Research and education in computational science and engineering. *Siam Review*, 60(3):707–754, 2018.

[3] John G Michopoulos, Charbel Farhat, and Jacob Fish. Modeling and simulation of multiphysics systems. *Journal of Computing and Information Science in Engineering*, 5(3):198–213, 2005.

[4] A.S. Lup. *Multiphysics Modelling of Radio Frequency Micro-Electro-Mechanical-Systems*. PhD thesis, University Politehnica of Bucharest, Romania, 2016.

[5] D. Ioan. Modelarea multifizică. ToMeMS - internal report - D1.1a "Modelarea cuplata electrostatica si mecanica a microcomutatoarelor de RF - Formularea problemei", 2012.

[6] D. Ioan. *Modelarea dispozitivelor electromagnetice*. UPB, Bucureşti, 2000. `http://www.lmn.pub.ro/~daniel/cursmde.pdf`.

[7] Alexander H-D Cheng and Daisy T Cheng. Heritage and early history of the boundary element method. *Engineering Analysis with Boundary Elements*, 29(3):268–302, 2005.

[8] A Singer, D Gillespie, J Norbury, and RS Eisenberg. Singular perturbation analysis of the steady-state poisson–nernst–planck system: Applications to ion channels. *European journal of applied mathematics*, 19(5):541–560, 2008.

[9] D. Ioan. *Metode pentru calculul campului electromagnetic. Separarea variabileleor*. Institutul Politehnic Bucuresti, Editura IPB, Bucureşti, 1988.

[10] A. Bossavit and I.D. Mayergoyz. *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*. Electromagnetism. Elsevier Science, 1998. `https://books.google.ro/books?id=RQ_9hc8U7MsC`.

[11] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, 2008. ISBN:9780980232707, `https://books.google.ro/books?id=K5MAOwAACAAJ`.

[12] D. Ioan. *Metoda elementului finit pentru modelarea electromagnetică*. UPB, Bucureşti, 2013.

[13] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. *Handbook of numerical analysis*, 7:713–1018, 2000.

[14] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[15] M Clemens T Weiland. Discrete electromagnetism with the finite integration technique. *Progress In Electromagnetics Research*, 32:65–87, 2001.

[16] René Marklein. The finite integration technique as a general tool to compute acoustic, electromagnetic, elastodynamic, and coupled wave fields. *Review of radio science*, 2002:201–244, 1999.

[17] Alain Bossavit and Lauri Kettunen. Yee-like schemes on staggered cellular grids: A synthesis between fit and fem approaches. *IEEE Transactions on Magnetics*, 36(4):861–867, 2000.

[18] John Volakis. *Integral equation methods for electromagnetics*. The Institution of Engineering and Technology, 2012.

[19] Gernot Beer. *Programming the boundary element method*. John Wiley & Sons, Inc., 2000.

[20] Thomas J Snowden, Piet H van der Graaf, and Marcus J Tindall. Methods of model reduction for large-scale biological systems: a survey of current methods and trends. *Bulletin of mathematical biology*, 79(7):1449–1486, 2017.

[21] Chen Li, Marco Donizelli, Nicolas Rodriguez, Harish Dharuri, Lukas Endler, Vijayalakshmi Chelliah, Lu Li, Enuo He, Arnaud Henry, Melanie I Stefan, et al. Biomodels database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC systems biology*, 4(1):92, 2010.

[22] Athanasios C Antoulas, Danny C Sorensen, and Serkan Gugercin. A survey of model reduction methods for large-scale systems. Technical report, 2000.

[23] Athanasios C Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Siam, 2005.

[24] Peter Benner, Roland W Freund, Danny C Sorensen, and Andras Varga. Special issue on âĂIJorder reduction of large-scale systemsâĂİ, 2006.

[25] Wilhelmus HA Schilders, Henk A Van der Vorst, and Joost Rommes. *Model order reduction: theory, research aspects and applications*, volume 13. Springer, 2008.

[26] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

[27] Ruxandra Bărbulescu and Daniel Ioan. Multi-scale and reduced modelling of biological systems. In *Advanced Topics in Electrical Engineering (ATEE), 2015 9th International Symposium on*, pages 597–602. IEEE, 2015.

[28] Hiroaki Kitano et al. *Foundations of systems biology*. MIT press Cambridge, 2001.

[29] Joseph Walpole, Jason A Papin, and Shayn M Peirce. Multiscale computational models of complex biological systems. *Annual review of biomedical engineering*, 15:137–154, 2013.

[30] Xinhao Ye, Ju Chu, Yinping Zhuang, and Siliang Zhang. Multi-scale methodology: a key to deciphering systems biology. *Front Biosci*, 10:961–965, 2005.

[31] Jorg Stelling, Andreas Kremling, Martin Ginkel, Katja Bettenbrock, and Ernst D Gilles. Towards a virtual biological laboratory. In *Foundations of systems biology*, pages 189–212. MIT Press, 2001.

[32] Thomas Simon Shimizu and Dennis Bray. 10 computational cell biologyâĂŤthe stochastic approach. *Foundations of Systems Biology*, page 213, 2001.

[33] Yoichi Nakayama and Masaru Tomita. 11 computer simulation of the cell: Human erythrocyte model and its application. *Foundations of Systems Biology*, page 233.

[34] Masaru Tomita, Kenta Hashimoto, Koichi Takahashi, Thomas Simon Shimizu, Yuri Matsuzaki, Fumihiko Miyoshi, Kanako Saito, Sakura Tanida, Katsuyuki Yugi, J Craig Venter, et al. E-cell: software environment for whole-cell simulation. *Bioinformatics (Oxford, England)*, 15(1):72–84, 1999.

[35] Rajiv Chaturvedi, Chengbang Huang, Bogdan Kazmierczak, T Schneider, Jesus A Izaguirre, Tilmann Glimm, H George E Hentschel, JA Glazier, SA Newman, and MS Alber. On multi-scale approaches to three-dimensional modelling of morphogenesis. *Journal of the Royal Society interface*, 2(3):237–253, 2005.

[36] James Southern, Joe Pitt-Francis, Jonathan Whiteley, Daniel Stokeley, Hiromichi Kobashi, Ross Nobes, Yoshimasa Kadooka, and David Gavaghan. Multi-scale computational modelling in biology and physiology. *Progress in biophysics and molecular biology*, 96(1-3):60–89, 2008.

[37] Matej Praprotnik, Luigi Delle Site, and Kurt Kremer. Multiscale simulation of soft matter: From scale bridging to adaptive resolution. *Annu. Rev. Phys. Chem.*, 59:545–571, 2008.

[38] Teemu Murtola, Alex Bunker, Ilpo Vattulainen, Markus Deserno, and Mikko Karttunen. Multiscale modeling of emergent materials: biological and soft matter. *Physical Chemistry Chemical Physics*, 11(12):1869–1892, 2009.

[39] Peter MA Sloot and Alfons G Hoekstra. Multi-scale modelling in computational biomedicine. *Briefings in bioinformatics*, 11(1):142–152, 2009.

[40] Joseph O Dada and Pedro Mendes. Multi-scale modelling and simulation in systems biology. *Integrative Biology*, 3(2):86–96, 2011.

[41] Shina Caroline Lynn Kamerlin and Arieh Warshel. Multiscale modeling of biological functions. *Physical Chemistry Chemical Physics*, 13(22):10401–10411, 2011.

[42] Zhilin Qu, Alan Garfinkel, James N Weiss, and Melissa Nivala. Multi-scale modeling in biology: how to bridge the gaps between scales? *Progress in biophysics and molecular biology*, 107(1):21–31, 2011.

[43] Monika Heiner and David Gilbert. Biomodel engineering for multiscale systems biology. *Progress in biophysics and molecular biology*, 111(2-3):119–128, 2013.

[44] Huan-Xiang Zhou. Theoretical frameworks for multiscale modeling and simulation. *Current opinion in structural biology*, 25:67–76, 2014.

[45] JP Wikswo, A Prokop, F Baudenbacher, D Cliffel, B Csukas, and M Velkovsky. Engineering challenges of bionems: the integration of microfluidics, micro-and nanodevices, models and external control for systems biology. In *Iee Proceedings-Nanobiotechnology*, volume 153, pages 81–101. IET, 2006.

[46] Ramon Grima. Multiscale modeling of biological pattern formation. *Current topics in developmental biology*, 81:435–460, 2008.

[47] Zoltán N Oltvai and Albert-László Barabási. Life's complexity pyramid. *Science*, 298(5594):763–764, 2002.

[48] E May and R Schiek. Xyce parallel electronic simulator: Biological pathway modeling and simulation, sand report sand2006-1993p, sandia national laboratories. Technical report, Tech. Rep, 2006.

[49] Lorant Janosi. *Multiscale modeling of biomolecular systems.* PhD thesis, University of Missouri–Columbia, 2007.

[50] ML Martins, SC Ferreira Jr, and MJ Vilela. Multiscale models for biological systems. *Current opinion in colloid & Interface Science*, 15(1-2):18–23, 2010.

[51] Amina A Qutub, Feilim Mac Gabhann, Emmanouil D Karagiannis, Prakash Vempati, and Aleksander S Popel. Multiscale models of angiogenesis. *IEEE engineering in medicine and biology magazine*, 28(2), 2009.

[52] Carsten Maus, Stefan Rybacki, and Adelinde M Uhrmacher. Rule-based multi-level modeling of cell biological systems. *BMC Systems Biology*, 5(1):166, 2011.

[53] Martin Meier-Schellersheim, Iain DC Fraser, and Frederick Klauschen. Multiscale modeling for biologists. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(1):4–14, 2009.

[54] Ovidiu Radulescu, Alexander N Gorban, Andrei Zinovyev, and Alain Lilienbaum. Robust simplifications of multiscale biochemical networks. *BMC systems biology*, 2(1):86, 2008.

[55] Alexander P Lyubartsev and Aatto Laaksonen. Calculation of effective interaction potentials from radial distribution functions: A reverse monte carlo approach. *Physical Review E*, 52(4):3730, 1995.

[56] Zhaojun Bai, Patrick M Dewilde, and Roland W Freund. Reduced-order modeling. *Handbook of numerical analysis*, 13:825–895, 2005.

[57] Dmitry Missiuro Vasilyev. *Theoretical and practical aspects of linear and nonlinear model order reduction techniques*. PhD thesis, Massachusetts Institute of Technology, 2008.

[58] Daniel Ioan, Gabriela Ciuprina, and Luis Miguel Silveira. Effective domain partitioning with electric and magnetic hooks. *IEEE Transactions on Magnetics*, 45(3):1328–1331, 2009.

[59] Gabriela Ciuprina, Daniel Ioan, Ioan Alexandru Lazar, and Cosmin Bogdan Dita. Vector fitting based adaptive frequency sampling for compact model extraction on hpc systems. *IEEE Transactions on Magnetics*, 48(2):431–434, 2012.

[60] Lorenzo Codecasa, Dario D'Amore, Paolo Maffezzoni, and Nadia Spennagallo. A novel approach for generating dynamic compact models of thermal networks having large numbers of power sources. In *THERMINIC 2005*, pages 25–31. TIMA Editions, 2005.

[61] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4. Springer Science & Business Media, 2009.

[62] Marc A Hesse, Bradley T Mallison, and Hamdi A Tchelepi. Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations. *Multiscale Modeling & Simulation*, 7(2):934–962, 2008.

[63] Victorita Dolean, Pierre Jolivet, Frédéric Nataf, Nicole Spillane, and Hua Xiang. Two-level domain decomposition methods for highly heterogeneous darcy equations. connections with multiscale methods. *Oil & Gas Science and Technology–Revue dâĂŹIFP Energies nouvelles*, 69(4):731–752, 2014.

[64] Daniel Ioan, Ruxandra Bărbulescu, Jean Ciurea, Gabriela Ciuprina, and Przemyslaw Syrek. Simple hierarchical models of the transcranial magnetic stimulation. In *Advanced Topics in Electrical Engineering (ATEE), 2017 10th International Symposium on*, pages 398–403. IEEE, 2017.

[65] AF Huxley and R Stämpeli. Evidence for saltatory conduction in peripheral myelinated nerve fibres. *The Journal of physiology*, 108(3):315–339, 1949.

[66] Daniel Ioan, Ruxandra Bărbulescu, Luis Miguel Silveira, and Gabriela Ciuprina. Reduced order models of myelinated axonal compartments. *Journal of Computational Neuroscience*, under review.

[67] Science Photo Library KTSDESIGN/Getty Images. Neurons and their role in the nervous system. the background of nerve system. Digital image.

[68] Manuel María Villapecellín-Cid, Francisco Medina, and Laura María Roa. Internodal myelinated segments: Delay and rgc time-domain green function model. *IEEE transactions on biomedical engineering*, 51(2):389–391, 2004.

[69] Marc C Ford, Olga Alexandrova, Lee Cossell, Annette Stange-Marten, James Sinclair, Conny Kopp-Scheinpflug, Michael Pecka, David Attwell, and Benedikt Grothe. Tuning of ranvier node and internode properties in myelinated axons to adjust action potential timing. *Nature communications*, 6:8073, 2015.

[70] Werner Van Geit, Erik De Schutter, and Pablo Achard. Automated neuron model optimization techniques: a review. *Biological cybernetics*, 99(4-5):241–251, 2008.

[71] F Tuisku and C Hildebrand. Nodes of ranvier and myelin sheath dimensions along exceptionally thin myelinated vertebrate pns axons. *Journal of neurocytology*, 21(11):796–806, 1992.

[72] Alexandru Timotin. La structure de la fibre nerveuse: Un projet optimal. *Proceedings Of The Romanian Academy, Series A*, 5(1), 2004.

[73] Wilfrid Rall. Core conductor theory and cable properties of neurons. *Comprehensive physiology*, pages 39–97, 2011.

[74] KA Lindsay, JM Ogden, DM Halliday, and JR Rosenberg. An introduction to the principles of neuronal modelling. In *Modern techniques in neuroscience research*, pages 213–306. Springer, 1999.

[75] Walter Freeman. *Neurodynamics: an exploration in mesoscopic brain dynamics*. Springer Science & Business Media, 2012.

[76] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience*, volume 806. Cambridge, MA: MIT Press, 2001.

[77] Wulfram Gerstner and Werner M. Kistler. Spiking neuron models. single neurons, populations, plasticity, 2002.

[78] Larry Squire, Darwin Berg, Floyd E Bloom, Sascha Du Lac, Anirvan Ghosh, and Nicholas C Spitzer. *Fundamental neuroscience*. Academic Press, 2012.

[79] John E Hall. *Guyton and Hall textbook of medical physiology e-Book*. Elsevier Health Sciences, 2015.

[80] Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

[81] James Keener and James Sneyd. *Mathematical physiology: I: cellular physiology*. Springer Science & Business Media, 2010.

[82] David Sterratt, Bruce Graham, Andrew Gillies, and David Willshaw. *Principles of computational modelling in neuroscience*. Cambridge University Press, 2011.

[83] J.M. Bower and D. Beeman. *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. Interpretation of Music; 55. TELOS, 1998.

[84] Nicholas T Carnevale and Michael L Hines. *The NEURON book*. Cambridge University Press, 2006.

[85] Frank Rattay, Thomas Potrusil, Cornelia Wenger, Andrew K Wise, Rudolf Glueckert, and Anneliese Schrott-Fischer. Impact of morphometry, myelinization and synaptic current strength on spike conduction in human and cat spiral ganglion neurons. *PLoS One*, 8(11):e79256, 2013.

[86] JW Moore, F Ramon, and RW Joyner. Axon voltage-clamp simulations. i. methods and tests. *Biophysical journal*, 15(1):11–24, 1975.

[87] The nervous system (structure and function) (nursing) part 1. Accessed: 2017-12-14.

[88] Wikipedia contributors. Schwann cell — Wikipedia, the free encyclopedia, 2017. [Online; accessed 25-August-2018].

[89] Robert G Young, Ann M Castelfranco, and Daniel K Hartline. The âĂIJlillie transitionâĂİ: models of the onset of saltatory conduction in myelinating axons. *Journal of computational neuroscience*, 34(3):533–546, 2013.

[90] M Ibrahim, AM Butt, and M Berry. Relationship between myelin sheath diameter and internodal length in axons of the anterior medullary velum of the adult rat. *Journal of the neurological sciences*, 133(1):119–127, 1995.

[91] WA Wesselink, J Holsheimer, and HBK Boom. A model of the electrical behaviour of myelinated sensory nerve fibres based on human data. *Medical & biological engineering & computing*, 37(2):228–235, 1999.

[92] John W Moore, Ronald W Joyner, Micheal H Brill, Stephen D Waxman, and Manuel Najar-Joa. Simulations of conduction in uniform myelinated fibers. relative sensitivity to changes in nodal and internodal parameters. *Biophysical journal*, 21(2):147–160, 1978.

[93] N Ganapathy and JW Clark. Extracellular currents and potentials of the active myelinated nerve fiber. *Biophysical journal*, 52(5):749–761, 1987.

[94] DI Stephanova. Myelin as longitudinal conductor: a multi-layered model of the myelinated human motor nerve fibre. *Biological cybernetics*, 84(4):301–308, 2001.

[95] Angus M Brown and Martine Hamann. Computational modeling of the effects of auditory nerve dysmyelination. *Frontiers in neuroanatomy*, 8, 2014.

[96] Saurabh Chaubey and Shikha J Goodwin. A unified frequency domain model to study the effect of demyelination on axonal conduction. *Biomedical engineering and computational biology*, 7:BECB–S38554, 2016.

[97] Walter J Freeman. Rational approximations in the complex plane for laplace transforms of transcendental linear operators. In *Neurodynamics: An Exploration in Mesoscopic Brain Dynamics*, pages 69–89. Springer, 2000.

[98] Alan L Hodgkin and Andrew F u. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[99] Richard FitzHugh. Mathematical models of excitation and propagation in nerve. *Biological engineering*, 1(9):1–85, 1969.

[100] B Frankenhaeuser and AF Huxley. The action potential in the myelinated nerve fibre of xenopus laevis as computed on the basis of voltage clamp data. *The Journal of Physiology*, 171(2):302–315, 1964.

[101] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[102] Richard Fitzhugh. Computation of impulse initiation and saltatory conduction in a myelinated nerve fiber. *Biophysical journal*, 2(1):11–21, 1962.

[103] Robert Young. *Mathematical Modeling of the Evolution and Development of Myelin*. PhD thesis, University of Hawaii at Manoa, 2015.

[104] Johannes J Struijk, Jan Holsheimer, Gerlof G van der Heide, and Herman BK Boom. Recruitment of dorsal column fibers in spinal cord stimulation: influence of collateral branching. *IEEE Transactions on Biomedical Engineering*, 39(9):903–912, 1992.

[105] William B Marks and Gerald E Loeb. Action currents, internodal potentials, and extracellular records of myelinated mammalian nerve fibers derived from node potentials. *Biophysical journal*, 16(6):655–668, 1976.

[106] Johannes Jan Struijk. The extracellular potential of a myelinated nerve fiber in an unbounded medium and in nerve cuff models. *Biophysical journal*, 72(6):2457–2469, 1997.

[107] Cameron C McIntyre and Warren M Grill. Extracellular stimulation of central neurons: influence of stimulus waveform and frequency on neuronal output. *Journal of neurophysiology*, 88(4):1592–1604, 2002.

[108] Claude Bédard, Helmut Kröger, and Alain Destexhe. Modeling extracellular field potentials and the frequency-filtering properties of extracellular space. *Biophysical journal*, 86(3):1829–1842, 2004.

[109] Carl Gold, Darrell A Henze, Christof Koch, and Gyorgy Buzsaki. On the origin of the extracellular action potential waveform: a modeling study. *Journal of neurophysiology*, 95(5):3113–3128, 2006.

[110] Paula Kuokkanen. *On the origin of the extracellular potential in the nucleus laminaris of the barn owl*. PhD thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2012.

[111] Sébastien Joucla, Alain Glière, and Blaise Yvert. Current approaches to model extracellular electrical neural microstimulation. *Frontiers in computational neuroscience*, 8, 2014.

[112] Harilal Parasuram, Bipin Nair, Egidio D'Angelo, Michael Hines, Giovanni Naldi, and Shyam Diwakar. Computational modeling of single neuron extracellular electric potentials and network local field potentials using lfpsim. *Frontiers in computational neuroscience*, 10, 2016.

[113] John Clark and Robert Plonsey. A mathematical evaluation of the core conductor model. *Biophysical journal*, 6(1):95–112, 1966.

[114] Anthony N Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95(1):1–19, 2006.

[115] François Chapeau-Blondeau and Nicolas Chambet. Synapse models for neural networks: from ion channel kinetics to multiplicative coefficient w ij. *Neural computation*, 7(4):713–734, 1995.

[116] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological cybernetics*, 98(6):459–478, 2008.

[117] Michael Graupner and Nicolas Brunel. Mechanisms of induction and maintenance of spike-timing dependent plasticity in biophysical synapse models. *Frontiers in Computational Neuroscience*, 4, 2010.

[118] Kevin Gurney. *An introduction to neural networks*. CRC press, 1997.

[119] S.S. Haykin. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999.

[120] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[121] Hélene Paugam-Moisy and Sander Bohte. Computing with spiking neuron networks. In *Handbook of natural computing*, pages 335–376. Springer, 2012.

[122] Digicortex. Accessed: 2017-12-14.

[123] Brain facts and figures. Accessed: 2017-12-14.

[124] PA Robinson, CJ Rennie, DL Rowe, SC O'Connor, and E Gordon. Multiscale brain modelling. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457):1043–1050, 2005.

[125] Ruxandra Bărbulescu, Daniel Ioan, and Jean Ciurea. Simple 1d models for neuro-signals transmission along axons. In *Electrical and Power Engineering (EPE), 2016 International Conference and Exposition on*, pages 313–319. IEEE, 2016.

[126] Al Timotin, Viorica Hortopan, A Ifrim, and Marius PREDA. *Lecţii de bazele electrotehnicii*. Editura Didactică şi Pedagogică, 1970.

[127] D. Ioan. Curs bazele electrotehnicii, 2012. Bachelor course held at University Politehnica of Bucharest (UPB).

[128] Camelia Gabriel, Sami Gabriel, and y E Corthout. The dielectric properties of biological tissues: I. literature survey. *Physics in Medicine & Biology*, 41(11):2231, 1996.

[129] Itis foundation tissue properties. dielectric properties.

[130] Italian national research council, dielectric properties of body tissues.

[131] Ichiji Tasaki. New measurements of the capacity and the resistance of the myelin sheath and the nodal membrane of the isolated frog nerve fiber. *American Journal of Physiology-Legacy Content*, 181(3):639–650, 1955.

[132] Wikipedia contributors. Cable theory — Wikipedia, the free encyclopedia, 2018. [Online; accessed 26-August-2018].

[133] Brown University. Brown engineering course, nernst potential and nerve conduction speed, 2007. Course Lectures, Brown University.

[134] Idan Segev. Cable and compartmental models of dendritic trees. In *The Book of Genesis*, pages 51–77. Springer, 1998.

[135] Rodney Cotterill. Biophysics: an introduction, 2003.

[136] Hyperphysics. Action potentials.

[137] PhysiologyWeb. Resting membrane potential.

[138] Wikipedia contributors. Resting potential — Wikipedia, the free encyclopedia, 2018. [Online; accessed 26-August-2018].

[139] Juraj Valsa. *Numerical Inversion of Laplace Transforms in Matlab*. The MathWorks Inc., Natick, Massachusetts, 2011.

[140] Wilfrid Rall. Rall model. *Scholarpedia*, 4(4):1369, 2009.

[141] Ernst Niebur. Neuronal cable theory. *Scholarpedia*, 3(5):2674, 2008.

[142] Michael Ohme and Andreas Schierwagen. An equivalent cable model for neuronal trees with active membrane. *Biological cybernetics*, 78(3):227–243, 1998.

[143] F Carlsen and F Behse. Three dimensional analysis of schwann cells associated with unmyelinated nerve fibres in human sural nerve. *Journal of anatomy*, 130(Pt 3):545, 1980.

[144] S Fred. Effects of demyelination on the time constants and amplitude of the action potential due to multiple sclerosis.

[145] Robert Wilkins, Simon Cross, Ian Megson, and David Meredith. *Oxford handbook of medical sciences*. OUP Oxford, 2011.

[146] Nathan A Angel. Equivalent circuit implementation of demyelinated human neuron in spice. 2011.

[147] Wikipedia contributors. Nerve conduction velocity — Wikipedia, the free encyclopedia, 2018. [Online; accessed 26-August-2018].

[148] Theodore Bullock and G Adrian Horridge. Structure and function in the nervous systems of invertebrates. 1965.

[149] OpenWetWare. Lab 9: Conduction velocity of nerves. fiber diameter.

[150] MJ Gillespie and RB Stein. The relationship between axon diameter, myelin thickness and conduction velocity during atrophy of mammalian peripheral nerves. *Brain research*, 259(1):41–56, 1983.

[151] Daniel Hartline. Evolution of pre-myelinate rapid conduction.

[152] Analog Devices Linear Technology. Ltspice.

[153] LTwiki. Lossy transmission line.

[154] Hans De Sterck and Paul Ullrich. Introduction to computational pdes. *Course Notes for Amath*, 442, 2009.

[155] Gerald W Recktenwald. Finite-difference approximations to the heat equation. *Mechanical Engineering*, 10:1–27, 2004.

[156] Rex Cheung. *Matlab script for digitizing a published graph*. The MathWorks Inc., Natick, Massachusetts, 2012.

[157] Katarina V Kolaric, Gemma Thomson, Julia M Edgar, and Angus M Brown. Focal axonal swellings and associated ultrastructural changes attenuate conduction velocity in central nervous system axons: a computer modeling study. *Physiological reports*, 1(3), 2013.

[158] KW Morton and DF Mayers. Numerical solution of partial differential equations. *Journal of Fluid Mechanics*, 363:349–349, 1998.

[159] MATLAB. *pdepe – Solve initial-boundary value problems for parabolic-elliptic PDEs in 1-D*. The MathWorks Inc., Natick, Massachusetts, 2015.

[160] Francesca Rapetti and Germain Rousseaux. On quasi-static models hidden in maxwell's equations. *Applied Numerical Mathematics*, 79:92–106, 2014.

[161] Daniel Ioan and Irina Munteanu. Missing link rediscovered: The electromagnetic circuit element concept. *JSAEM Studies in Applied Electromagnetics and Mechanics*, 8:302–320, 1999.

[162] Jian-Ming Jin. *The finite element method in electromagnetics*. John Wiley & Sons, 2015.

[163] Daniel Ioan, Gabriela Ciuprina, Marius Radulescu, and Ehrenfried Seebacher. Compact modeling and fast simulation of on-chip interconnect lines. *IEEE Transactions on Magnetics*, 42(4):547–550, 2006.

[164] Bjorn Gustavsen and Adam Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on power delivery*, 14(3):1052–1061, 1999.

[165] The Vector Fitting Web Site. Vector fitting, 2013.

[166] Gabriela Ciuprina, Jorge Fernández Villena, Daniel Ioan, Zoran Ilievski, Sebastian Kula, E Jan W ter Maten, Kasra Mohaghegh, Roland Pulch, Wil HA Schilders, L Miguel Silveira, et al. Parameterized model order reduction. In *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, pages 267–359. Springer, 2015.

[167] Petre Stoica, Randolph L Moses, et al. *Spectral analysis of signals*, volume 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

[168] Milton Abramowitz, Irene A Stegun, and Robert H Romer. Handbook of mathematical functions with formulas, graphs, and mathematical tables, 1988.

[169] Daniel W Lozier. Nist digital library of mathematical functions. *Annals of Mathematics and Artificial Intelligence*, 38(1):105–119, 2003.

[170] PJ Pieter Heres. *Robust and efficient Krylov subspace methods for model order reduction*. PhD thesis, Technische Universiteit Eindhoven, 2005.

[171] Heiko Panzer. *Model order reduction by Krylov subspace methods with global error bounds and automatic choice of parameters*. Verlag Dr. Hut, 2014.

[172] MORwiki. Balanced truncation, 2018. [Online; accessed 29-August-2018].

[173] MATLAB. *balancmr – Balanced model truncation via square root method*. The MathWorks Inc., Natick, Massachusetts, 2015.

[174] MATLAB. *balred – Model order reduction*. The MathWorks Inc., Natick, Massachusetts, 2015.

[175] MATLAB. *modred – Eliminate states from state-space models*. The MathWorks Inc., Natick, Massachusetts, 2015.

[176] A. Megretski. 6.243j dynamics of nonlinear systems. fall 2003., 2003. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

[177] Serkan Gugercin and Athanasios C Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.

[178] Joel R Phillips and Luís Miguel Silveira. Poor man's tbr: a simple model reduction scheme. *IEEE transactions on computer-aided design of integrated circuits and systems*, 24(1):43–55, 2005.

[179] Behnam Salimbahrami and Boris Lohmann. Krylov subspace methods in linear model order reduction: Introduction and invariance properties. In *Sci. Rep.. Inst. of Automation*. Citeseer, 2002.

[180] Mario Berljafa and Stefan Güttel. A rational krylov toolbox for matlab. 2014.

[181] Ilse CF Ipsen and Carl D Meyer. The idea behind krylov methods. *The American mathematical monthly*, 105(10):889–899, 1998.

[182] Zhaojun Bai and Roland W Freund. A partial padé-via-lanczos method for reduced-order modeling. *Linear Algebra and its Applications*, 332:139–164, 2001.

[183] Eric James Grimme. *Krylov projection methods for model reduction*. PhD thesis, Citeseer, 1997.

[184] Imad M Jaimoukha and Ebrahim M Kasenally. Implicitly restarted krylov subspace methods for stable partial realizations. *SIAM Journal on Matrix Analysis and Applications*, 18(3):633–652, 1997.

[185] Zhaojun Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied numerical mathematics*, 43(1-2):9–44, 2002.

[186] Roland W Freund. Krylov-subspace iterations for reduced-order modeling in vlsi circuit simulation. Technical report, Citeseer.

[187] Roland W Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123(1-2):395–421, 2000.

[188] Stefan Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz. see http://www. uni-graz. at/imawww/volkwein/POD. pdf*, 1025, 2011.

[189] Karen Willcox and Jaime Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA journal*, 40(11):2323–2330, 2002.

[190] MATLAB. *modred – Singular value decomposition*. The MathWorks Inc., Natick, Massachusetts, 2015.

[191] William C Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of applied physics*, 19(1):55–63, 1948.

[192] John Robert Burger. *Human memory modeled with standard analog and digital circuits: inspiration for man-made computers*. John Wiley & Sons, 2009.

[193] Ruxandra Bărbulescu, Daniel Ioan, and Gabriela Ciuprina. Modeling the saltatory conduction in myelinated axons by order reduction. In *Neuroinformatics and Computational Neuroscience (ICNCN), 2018 20th International Conference on*, pages 2059–2062, 2018.

[194] Ichiji Tasaki. *Physiology and electrochemistry of nerve fibers*. Elsevier, 2012.

[195] Charles F Babbs and Riyi Shi. Subtle paranodal injury slows impulse conduction in a mathematical model of myelinated axons. *PLoS One*, 8(7):e67767, 2013.

[196] COMSOL Multiphysics®. *Simulating Action Potential with the Hodgkin-Huxley Model*. COMSOL AB, Stockholm, Sweden. Demo App.

[197] EECS Department of the University of California at Berkeley. The spice page.

[198] S Grissmer. Properties of potassium and sodium channels in frog internode. *The Journal of physiology*, 381(1):119–134, 1986.

[199] Nele De Geeter, Guillaume Crevecoeur, and Luc Dupré. An efficient 3-d eddy-current solver using an independent impedance method for transcranial magnetic stimulation. *IEEE Transactions on Biomedical Engineering*, 58(2):310–320, 2011.

[200] Serena Fiocchi, Michela Longhi, Paolo Ravazzani, Yiftach Roth, Abraham Zangen, and Marta Parazzini. Modelling of the electric field distribution in deep transcranial magnetic stimulation in the adolescence, in the adulthood, and in the old age. *Computational and mathematical methods in medicine*, 2016, 2016.

[201] Simone Rossi, Mark Hallett, Paolo M Rossini, Alvaro Pascual-Leone, Safety of TMS Consensus Group, et al. Safety, ethical considerations, and application guidelines for the use of transcranial magnetic stimulation in clinical practice and research. *Clinical neurophysiology*, 120(12):2008–2039, 2009.

[202] Masahito Kobayashi and Alvaro Pascual-Leone. Transcranial magnetic stimulation. In *The Clinical Neurophysiology Primer*, pages 499–515. Springer, 2007.

[203] Risto J Ilmoniemi, Juha Virtanen, Jarmo Ruohonen, Jari Karhu, Hannu J Aronen, Risto Näätänen, and Toivo Katila. Neuronal responses to magnetic stimulation reveal cortical reactivity and connectivity. *Neuroreport*, 8(16):3537–3540, 1997.

[204] Vanessa Guadagnin, Marta Parazzini, Serena Fiocchi, Ilaria Liorni, and Paolo Ravazzani. Deep transcranial magnetic stimulation: Modeling of different coil configurations. *IEEE Trans. Biomed. Engineering*, 63(7):1543–1550, 2016.

[205] Anthony T Barker, Reza Jalinous, and Ian L Freeston. Non-invasive magnetic stimulation of human motor cortex. *The Lancet*, 325(8437):1106–1107, 1985.

[206] Zhi-De Deng, Sarah H Lisanby, and Angel V Peterchev. Electric field depth–focality tradeoff in transcranial magnetic stimulation: simulation comparison of 50 coil designs. *Brain stimulation*, 6(1):1–13, 2013.

[207] Joshua M Saypol, Bradley J Roth, Leonardo G Cohen, and Mark Hallett. A theoretical comparison of electric and magnetic stimulation of the brain. *Annals of biomedical engineering*, 19(3):317–328, 1991.

[208] R Salvador, S Silva, PJ Basser, and PC Miranda. Determining which mechanisms lead to activation in the motor cortex: a modeling study of transcranial magnetic stimulation using realistic stimulus waveforms and sulcal geometry. *Clinical neurophysiology*, 122(4):748–758, 2011.

[209] Abraham Zangen, Yiftach Roth, Bernhard Voller, and Mark Hallett. Transcranial magnetic stimulation of deep brain regions: evidence for efficacy of the h-coil. *Clinical neurophysiology*, 116(4):775–779, 2005.

[210] H Eaton. Electric field induced in a spherical volume conductor from arbitrary coils: application to magnetic stimulation and meg. *Medical and Biological Engineering and Computing*, 30(4):433–440, 1992.

[211] Bradley J Roth, Leonardo G Cohen, and Mark Hallett. The electric field induced during magnetic stimulation. *Electroencephalography and Clinical Neurophysiology-Supplements only*, 43:268–278, 1994.

[212] FS Salinas, JL Lancaster, and PT Fox. 3d modeling of the total electric field induced by transcranial magnetic stimulation using the boundary element method. *Physics in Medicine & Biology*, 54(12):3631, 2009.

[213] Steve K Esser, Sean L Hill, and Giulio Tononi. Modeling the effects of transcranial magnetic stimulation on cortical circuits. *Journal of neurophysiology*, 94(1):622–639, 2005.

[214] T Kowalski, J Silny, and H Buchner. Current density threshold for the stimulation of neurons in the motor cortex area. *Bioelectromagnetics: Journal of the Bioelectromagnetics Society, The Society for Physical Regulation in Biology and Medicine, The European Bioelectromagnetics Association*, 23(6):421–428, 2002.

[215] Aapo Nummenmaa, Matti Stenroos, Risto J Ilmoniemi, Yoshio C Okada, Matti S Hämäläinen, and Tommi Raij. Comparison of spherical and realistically shaped boundary element head models for transcranial magnetic stimulation navigation. *Clinical Neurophysiology*, 124(10):1995–2007, 2013.

[216] Charles M Epstein, Daniel G Schwartzberg, Kent R Davey, and David B Sudderth. Localizing the site of magnetic brain stimulation in humans. *Neurology*, 40(4):666–666, 1990.

[217] Przemyslaw Syrek and Ruxandra Bărbulescu. Parametric curves to trace the tms coils windings. In *Advanced Topics in Electrical Engineering (ATEE), 2017 10th International Symposium on*, pages 386–391. IEEE, 2017.

[218] Martin J Mohlenkamp. A userâĂŹs guide to spherical harmonics, 2010.

[219] James C Simpson, John E Lane, Christopher D Immer, and Robert C Youngquist. Simple analytic expressions for the magnetic field of a circular current loop. 2001.

[220] Jitesh Jain, Cheng-Kok Koh, and Venkataramanan Balakrishnan. Exact and numerically stable closed-form expressions for potential coefficients of rectangular conductors. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(6):458–462, 2006.

[221] Henry C Thacher Jr. Remark on algorithm 60: Romberg integration. *Communications of the ACM*, 7(7):420–421, 1964.

[222] Vyacheslav Ivanovich Lebedev and DN Laikov. A quadrature formula for the sphere of the 131st algebraic order of accuracy. In *Doklady Mathematics*, volume 59, pages 477–481. Pleiades Publishing, Ltd.(Đ§ĐżĐţĐřĐťĐţÑĄ Đ§ĐřĐśĐżĐÿÑĹĐÿĐ¡Đş, ĐŻÑĆĐť), 1999.

[223] e MRI. Mri step-by-step, interactive course on magnetic resonance imaging, 2018.

[224] R Salvador, A Mekonnen, G Ruffini, and PC Miranda. Modeling the electric field induced in a high resolution realistic head model during transcranial current stimulation. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2073–2076. IEEE, 2010.

[225] McGill University Montréal Canada McConnell Brain Imaging Centre, Montréal Neurological Institute. Brainweb: Simulated brain database.

[226] Chris A Cocosco, Vasken Kollokian, Remi K-S Kwan, G Bruce Pike, and Alan C Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*. Citeseer, 1997.

[227] David W Shattuck and Richard M Leahy. Brainsuite: an automated cortical surface identification tool. *Medical image analysis*, 6(2):129–142, 2002.

[228] Won Hee Lee, Zhi-De Deng, Tae-Seong Kim, Andrew F Laine, Sarah H Lisanby, and Angel V Peterchev. Regional electric field induced by electroconvulsive therapy in a realistic finite element head model: influence of white matter anisotropic conductivity. *Neuroimage*, 59(3):2110–2123, 2012.

[229] Karl J Friston. Introduction: experimental design and statistical parametric mapping. *Human brain function*, 2004.

[230] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.12 edition, 2018.

[231] Pedro Cavaleiro Miranda, Mark Hallett, and Peter J Basser. The electric field induced in the brain by magnetic stimulation: a 3-d finite-element analysis of the effect of tissue heterogeneity and anisotropy. *IEEE transactions on biomedical engineering*, 50(9):1074–1085, 2003.

[232] Tim A Wagner, Markus Zahn, Alan J Grodzinsky, and Alvaro Pascual-Leone. Three-dimensional head model simulation of transcranial magnetic stimulation. *IEEE Transactions on Biomedical Engineering*, 51(9):1586–1598, 2004.

[233] Anders M Dale, Bruce Fischl, and Martin I Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, 1999.

[234] Andreas Christ, Wolfgang Kainz, Eckhart G Hahn, Katharina Honegger, Marcel Zefferer, Esra Neufeld, Wolfgang Rascher, Rolf Janka, Werner Bautz, Ji Chen, et al. The virtual familyâĂŤdevelopment of surface-based anatomical models of two adults and two children for dosimetric simulations. *Physics in Medicine & Biology*, 55(2):N23, 2009.

[235] C Gabriel, A Peyman, and EH Grant. Electrical conductivity of tissue at frequencies below 1 mhz. *Physics in medicine & biology*, 54(16):4863, 2009.

[236] Bin Yang, Kwong-Ming Tse, Ning Chen, Long-Bin Tan, Qing-Qian Zheng, Hui-Min Yang, Min Hu, Gang Pan, and Heow-Pueh Lee. Development of a finite element head model for the study of impact head injury. *BioMed research international*, 2014, 2014.

[237] Bruce Fischl, David H Salat, Evelina Busa, Marilyn Albert, Megan Dieterich, Christian Haselgrove, Andre Van Der Kouwe, Ron Killiany, David Kennedy, Shuna Klaveness, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355, 2002.

[238] Frank H Netter. *Atlas of Human Anatomy E-Book*. Elsevier Health Sciences, 2017.

[239] Faris Tarlochan et al. Finite element (fe) human head models/literature review. *International Journal of Scientific & Technology Research*, 2(7):17–31, 2013.

[240] Egill A Friðgeirsson, Paolo Gargiulo, Ceon Ramon, and Jens Haueisen. 3d segmented model of head for modelling electrical activity of brain. *European Journal of Translational Myology*, 22(1-2):57–60, 2012.

[241] Ceon Ramon, Paul H Schimpf, and Jens Haueisen. Influence of head models on eeg simulations and inverse source localizations. *BioMedical Engineering Online*, 5(1):10, 2006.

[242] Nandini Shrinidhi, David R Haynor, Yanqun Wang, Dawn B Jorgenson, Gust H Bardy, and Yongmin Kim. An efficient tissue classifier for building patient-specific finite element models from x-ray ct images. *IEEE transactions on biomedical engineering*, 43(3):333–337, 1996.

[243] Janakinadh Yanamadala. Development of human body cad models and related mesh processing algorithms with applications in bioelectromagnetics. 2016.

[244] Suyash P Awate. *Adaptive, nonparametric markov models and information-theoretic methods for image restoration and segmentation*. PhD thesis, School of Computing, University of Utah, 2006.

[245] Guido Gerig. Itk-snap user training, manual capabilities, presentation slides.

[246] Ivana Despotović, Bart Goossens, and Wilfried Philips. Mri segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015, 2015.

[247] K Kazemi and N Noorizadeh. Quantitative comparison of spm, fsl, and brainsuite for brain mr image segmentation. *Journal of biomedical physics & engineering*, 4(1):13, 2014.

[248] Qianqian Fang. iso2mesh: a 3d surface and volumetric mesh generator for matlab/octave.

[249] Qianqian Fang and David A Boas. Tetrahedral mesh generation from volumetric binary and grayscale images. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 1142–1145. Ieee, 2009.

[250] Qianqian Fang. Mesh-based monte carlo method using fast ray-tracing in plücker coordinates. *Biomedical optics express*, 1(1):165–175, 2010.

[251] D Louis Collins, Alex P Zijdenbos, Vasken Kollokian, John G Sled, Noor J Kabani, Colin J Holmes, and Alan C Evans. Design and construction of a realistic digital brain phantom. *IEEE transactions on medical imaging*, 17(3):463–468, 1998.

[252] COMSOL Multiphysics. *COMSOL – Supported File Formats*. COMSOL AB, Stockholm, Sweden.

[253] COMSOL Multiphysics. *Can COMSOL import topological data, meshes and images?* COMSOL AB, Stockholm, Sweden.

[254] Pau Micó. *stlTools: a collection of functions, samples and demos to illustrate how to deal with STL files*. The MathWorks Inc., Natick, Massachusetts, 2011.