Thomas Grandits

Optimization and Machine Learning for Inverse Problems in Cardiac Electrophysiology

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Prof. Thomas Pock
Institute of Computer Graphics and Vision, Graz University of Technology

Examiner

Prof. Alessandro Veneziani
Department of Mathematics, Emory College of Arts and Sciences

Graz, Austria, June 2021

English Abstract The heart is a fascinating and vital organ that has sparked interest among scholars as early as the antique. Its reliable functionality under drastically varying circumstances has not ceased to amaze scientists up to this day. Modern computational power allows unprecedented insights into the electromechanical cardiac functionality, but the topic remains a challenging field across many engineering disciplines, including mechanics, fluid structure interactions, as well as electrophysiology, despite considerable progress over the past decade. Still, no matter how sophisticated such high-fidelity models become, correct parameter selection is indispensable when trying to explain phenomena and pathologies, often encountered in the clinic. The study of parameter selection for said models also bears significance for academic purposes to better understand how the heart works.

This thesis builds on works of the past few years, dealing with physiological faithfully and mathematically sound ways to estimate parameters of existing and well-established partial differential equations. These equations have shown the ability to model measurements, encountered in simulations, but we also show some examples of models directly fitted onto real world measurements, such as endocardial electrical measurements during catheter ablation therapy.

Our underlying model assumption is the anisotropic eikonal equation, which has proven to be an efficient yet simple enough tool to model the electrical activation throughout the heart. While the eikonal equation still offers some computational hurdles with respect to inverse problems, we show how we can facilitate it to fit several of the most important electrophysiological parameters.

All of the presented methods and their results show that they may provide a good additional computational tool and guide for decision making of cardiologists in the future.

German Abstract Das Herz ist ein faszinierendes und essenzielles Organ, welches das Interesse von Wissenschaftlern seit der Antike erweckt hat. Die zuverlässige Funktionsweise des Herzmuskels, selbst unter sich stark ändernden Bedingungen, versetzt Forscher noch heute in Staunen. Modernste Technologien und daran gekoppelte Rechenleistung erlaubt noch nie dagewesene Einblicke in die elektromechanische Funktionalität des Herzens, stellt aber hohe Anforderung an diverse mathematische und Ingenieursdisziplinen, einschließlich Mechanik, Fluid-Struktur-Kopplung und Elektrophysiologie. Doch egal wie präzise und genau die Lösungsmethoden für die genannten Systeme werden, ihre Parameterschätzung ist unentbehrlich um auf täglicher Basis diagnostizierte Pathologien zu erklären. Erkentnisse über Parameter und deren Zusammenhänge bergen auch hohes Potenzial für weitere akademische Studien um das Herz besser zu verstehen.

Die vorliegende Arbeit baut auf Publikationen vergangener Jahre auf die sich mit physiologisch getreuen und mathematisch fundierten Methoden zur Parameterschätzung der im Feld etablierten partiellen Differentialgleichungen beschäftigen. Die genannten Gleichungen zeigten die Fähigkeit sowohl simulierte, als auch reale Messungen zu replizieren, so wie endokardiale, elektrische Messungen während elektrophysiologischer Untersuchun-

gen.

Unsere Modellannahme ist die anisotrope Eikonal Gleichung, welche sich als effizientes und naturgetreues Werkzeug zur Simulierung der elektrischen Herzaktivität bewiesen hat. Obwohl die Eikonal Gleichung trotzdem gewisse Herausforderungen an inverse Probleme stellt, zeigen wir, wie sie effektiv zur Schätzung wichtiger elektrophysiologischer Parameter genutzt werden kann.

Die hier vorgestellten Methoden und Ergebnisse gewähren einen Einblick in die Möglichkeiten potenzieller, zukünftiger Werkzeuge der Kardiologen zur Analyse des Herzens.

Acknowledgments

As this work is intended to conclude my PhD research, I wanted to take the time to thank all people that helped and accompanied me during this long endeavor. Special thanks goes here to the people of my private life, especially my friends that have supported me up to this day and helped me along way in achieving this daunting goal. A special shout-out to Karin, Fruela, Ulrike, and Tiago among others that have accompanied me on this journey. Special mentions are also due here to my family, specifically my siblings Philipp, Christina, my parents Franz, Christine and our newest member, Rafael.

Also, I'd also like to take the opportunity to thank all my colleagues at the Institute of Computer Graphics and Vision, Medical University, as well as all the people from external institutions that I had the pleasure to work with. You made the day-to-day work an enjoyable experience in an otherwise very competitive und unforgiving field and I wholeheartedly enjoyed discussions about all conceivable topics, especially with my office colleagues Thomas and Markus which joined me in my struggle to bear the fiery furnace that was our office up until a few years ago. A special mention needs to be had here for all the excellent people of the VLO group that helped me in better understanding the field of optimization and machine learning that has fascinated me since my Master's programme. It's my firm believe that you are doing a great job in methodologically advancing the different fields that you are working on and I hope this field of research and especially the group itself has a long and bright future.

I also wanted to thank the Center of Computational Medicine in Cardiology (CCMC[1]) in Lugano, Rolf and all the people there, who hosted me in the second half of 2019, as well as the Cardiocentro di Ticino with the very open minded Angelo as director of the clinical electrophysiology unit. Many thanks to all my colleagues there (Tobi, Hardik, Elena, Cyrill, ...) for interesting discussions and the leisure time spent together. A special thanks here goes to Simone with whom I had the pleasure to discuss many different approaches and directions for the field that engage me up until today and brought many research papers already to fruition. I thoroughly enjoyed our many discussions and methodologies that we developed together.

Please note that throughout the paper, the collective nouns, i.e. "we/us" will be used. This is neither an editorial error, nor an early sign of a multiple personality disorder. Rather, I'd like to acknowledge that this thesis would have not been possible, if it had not been for the collaboration, help, assistance and guidance I received by my supervisor

[1] `https://www.ccmc.usi.ch/`

Thomas Pock and my co-authors that made this work possible. On the one hand, they helped in fuelling my research output, by giving many useful pointers and hints that lead me to paths I could not have conceived myself, but on the other hand the critical feedback that tremendously helped in improving the perceived drafts and finished works. Also Erich and Alex deserve more than my gratitude for countless corrections of the presented work. Their continued effort helped in ensuring that the presented work is hopefully sound and an enjoyable reading experience.

Contents

Foreword 1

1 Introduction & Motivation 3
1.1 Structure of the Work 6
1.2 Discussed Publications 7

2 Mathematical Preliminaries 9
2.1 Notation 9
2.2 Functional Analysis 10
2.3 Matrices & Metrics 11
2.4 Continuous Functions 13
2.5 Non-continuous Functions 14
2.6 Ordinary Differential Equations 16
2.7 Differential Geometry 18
2.8 Partial Differential Equations (PDEs) 23
2.8.1 The Eikonal Equation 24
2.8.2 Discretization 27
2.9 Finite Element Method (FEM) 29
2.10 Optimization 34
2.10.1 Convex Optimization Problems 35
2.10.2 Non-smooth Optimization 38
2.10.3 Inverse Problems 41

3 Physiological Foundations 47
3.1 From chemical elements to electrical activation 47
3.2 Anatomy & ECG 54
3.3 Lead Fields 57

4 Proposed Methods 61
4.1 Related Methods 61
4.2 Methodological Foundations 64
4.3 Solving the Anisotropic Eikonal Equation - The Fast Iterative Method (Fast Iterative Method (FIM)) 65
4.4 Runtime 69

4.5 FIMIN: A direct approach at solver-based gradient computation 70
4.5.1 Inverse Problem - Gradient Computation 70
4.5.2 Optimization 72
4.5.3 Benchmarks 75
4.5.3.1 2D tissue sheet 76
4.5.3.2 Human biventricular model 76
4.5.3.3 Robustness 78
4.5.4 Results 79
4.5.4.1 2D-Sheet 79
4.5.4.2 Biventricular Models 79
4.5.4.3 Robustness 84
4.5.5 Discussion 85
4.5.6 Identifying the ventricular activation sequence 86
4.5.7 Robustness 88
4.5.8 Relation to inverse ECG imaging problems 89
4.5.9 Importance of ε 89
4.5.10 Computational Costs 89
4.5.11 Limitations 90
4.6 Endocardial contact mapping 91
4.6.1 Representation of the conductivity tensor 92
4.6.2 PIEMAP: Personalized Inverse Eikonal Model from cardiac Electro-Anatomical Maps 93
4.6.2.1 Methods 94
4.6.2.2 Experiments 96
4.6.2.3 Extended Evaluation on Clinical Data 99
4.6.3 Anisotropic Eikonal Net 100
4.6.3.1 Methods 102
4.6.3.2 Numerical experiments 105
4.7 Identifying cardiac initiation sites through geodesics - GEASI 106
4.7.1 The GEASI Method 108
4.7.1.1 Eikonal equation 108
4.7.1.2 Objective functional 110
4.7.1.3 Exponential Map 111
4.7.1.4 GEASI Algorithm 113
4.7.2 Extensions of GEASI 115
4.7.2.1 Variable number of EASs: Topological Gradient 116
4.7.2.2 Optimization using the ECG 117
4.7.3 Discretization 122
4.7.3.1 Solving the Eikonal Equation 122
4.7.3.2 Computation of Geodesics 122
4.7.3.3 Update of $\mathcal{E}^{(k+1)}$ 123

4.7.4 Numerical Results . . . 124
4.7.4.1 Activation Time Optimization . . . 125
4.7.4.2 Topological Gradient . . . 127
4.7.4.3 ECG . . . 130
4.7.5 Runtime . . . 132

5 Discussion & Conclusion 137
5.1 Eikonal Equation . . . 138
5.2 ECG . . . 139

Bibliography 141
Bibliography . . . 141

Appendix A FIMIN 155
A.1 Additional 2D Experiments . . . 155
A.2 Optimal Choice of λ . . . 156
A.3 Gradient of λ . . . 160
A.4 Gradient-Approximation of the Diffusion Tensors . . . 161

Appendix B List of Publications 163

List of Figures

1.1 Electrophysiological Schematic . . . 5

2.1 Tensor representation . . . 13
2.2 Sigmoid function . . . 18
2.3 Differentiable Manifold . . . 19
2.4 Tangent Space . . . 21
2.5 Manifold Example . . . 23
2.6 Weak solutions . . . 25
2.7 Vanishing Viscosity . . . 26
2.8 Finite Difference Grid . . . 29
2.9 Finite Element Method (FEM) Mapping . . . 31
2.10 The Periodic Table of Finite Elements (FEs) . . . 32
2.11 Softshrinkage . . . 39
2.12 Huber Norm . . . 42
2.13 Proximal Mapping of the Huber Norm . . . 44

3.1 Equilibrium State of chemical Concentrations in Cells . . . 48
3.2 Cell Action Potential Cycle . . . 49
3.3 Electrical Circuit Diagram of a single Cell Membrane . . . 51
3.4 Mitchell-Schaeffer Action Potential . . . 51
3.5 The Cable Equation . . . 52
3.6 Heart Anatomy . . . 55
3.7 12 Lead Electrocardiogram (ECG) . . . 56
3.8 Schematic ECG Representation . . . 57
3.9 Torso Lead Field . . . 60

4.1 Local FIM Update . . . 66
4.2 Floating Point Operations per Second (FLOPs/s) growth over the years . . . 69
4.3 FIM Updates . . . 72
4.4 FIMIN 2D Results . . . 80
4.5 FIMIN Biventricular Setup . . . 81
4.6 FIMIN Experiment Evaluation . . . 82
4.7 FIMIN Experiment Evaluation . . . 83
4.8 Distribution of Errors using FIMIN . . . 84
4.9 Distribution of optimized Velocities of FIMIN . . . 84
4.10 Distribution of Earliest Activation Site (EAS) Timings using FIMIN . . . 85
4.11 Energy Landscape of FIM for different ε . . . 90
4.12 Vector Heat Results . . . 94
4.13 Local Generated Bases of the Atria . . . 95

4.14 Results of PIEMAP on the in-silico model . . . 98
4.15 Results of PIEMAP on the in-vivo model . . . 100
4.16 Cross-validation study of PIEMAP . . . 101
4.17 Qualitative evaluation of PIEMAP . . . 103
4.18 Employed Physics Informed Neural Network (PINN) architecture . . . 104
4.19 PINN Results . . . 106
4.20 Eikonal and an associated Geodesic . . . 109
4.21 Visualization of the GEASI method . . . 114
4.22 Asymptotic Change w.r.t. EAS location . . . 117
4.23 Membrane Waveform . . . 120
4.24 Geodesic Discretization Accuracy . . . 123
4.25 GEASI Experiment Setups . . . 126
4.26 GEASI Results in 2D . . . 127
4.27 GEASI Results for in-vivo Data . . . 128
4.28 GEASI Results for an in-silico Model . . . 129
4.29 GEASI Results for the Topological Gradient #1 . . . 130
4.30 GEASI Results for the Topological Gradient #2 . . . 131
4.31 ECG Experiment Setups . . . 132
4.32 ECG Experiment Results in 2D . . . 133
4.33 ECG Experiment Results in 3D . . . 134
4.34 Geodesic Ordinary Differential Equation (ODE) Convergence of GEASI . . . 135

A.11 FIMIN Extra Experiment #1 . . . 155
A.12 FIMIN Extra Experiment #2 . . . 156

List of Tables

3.1 Chemical Concentration in Cells 50

4.1 FIMIN Results 86
4.2 Comparison ofPIEMAP with two other methods 97
4.3 Data Analysis of the Extended Evaluation 102
4.4 Extended Clinical Evaluation of PIEMAP 102
4.5 Evaluation of our PINN approach 106
4.6 GEASI Transmembrane Potential Parameters 120
4.7 GEASI Setups 125

List of Acronyms

ADAM	Adaptive Moment Estimation. 63, 139
AF	Atrial Fibrillation. 3, 54, 91, 92
AL	Active List. 67, 68
ANN	Artificial Neural Network. 8, 101, 137
APD	Action Potential Duration. 50, 56
BPSM	Body Potential Surface Map. 62, 107
CT	Computer Tomography. 91
DOF	Degrees of Freedom. 30
EAM	Electro-Anatomical Map. 7, 8, 61, 91, 99, 101, 102
EAS	Earliest Activation Site. vi, vii, 8, 64, 85, 106–109, 111, 113–117, 121–123, 125–135, 137
ECG	Electrocardiogram. vi, vii, 3, 4, 6, 47, 54–57, 60, 62, 107, 110, 115, 117, 119–121, 125, 130–135, 137–139
ECGi	Electrocardiographic Imaging. 62, 89
FD	Finite Difference. 28, 29
FDM	Finite Difference Method. 28, 29
FE	Finite Element. vi, 29–31, 33
FEM	Finite Element Method. vi, 6, 28, 29, 31, 34, 100, 101
FIM	Fast Iterative Method. iii, vi, 54, 61, 63, 65, 67, 68, 70–72, 92–94, 137
FISTA	Fast Iterative Shrinkage and Thresholding Algorithm. 44, 95, 137
FLOPs/s	Floating Point Operations per Second. vi, 69
FMM	Fast Marching Method. 54, 63
GPGPU	General Purpose Graphics Processing Unit. 4, 69
GPU	Graphics Processing Unit. 63, 69, 95
GT	Ground-Truth. 97, 98
HPC	High Performance Computing. 69
HPS	His-Purkinje system. 4, 5, 54
ISTA	Iterative Shrinkage and Thresholding Algorithm. 43, 44, 124
L-BFGS	Limited memory Broyden Fletcher Goldfarb Shanno. 46, 74, 124
LA	Left Atrium. 4, 54, 96, 97
LASSO	Least Absolute Shrinkage and Selection Operator. 43
LAT	Local Activation Time. 7, 62, 100, 103, 137
LBBB	Left Bundle Branch Block. 54
LV	Left Ventricle. 56
ML	Machine Learning. 5, 6, 17, 18, 63, 69, 94, 101, 104, 138, 139
MRI	Magnetic Resonance Imaging. 61
ODE	Ordinary Differential Equation. vii, 16, 17, 50, 69, 133, 135

PDE	Partial Differential Equation. 4–6, 8, 22–24, 27, 29, 33, 45, 50, 54, 58, 59, 62, 63, 69, 100–102, 138
PINN	Physics Informed Neural Network. vii, viii, 8, 63, 96, 100–102, 104–106, 138
PVJ	Purkinje ventricular junction. 76, 77
RA	Right Atrium. 4, 54, 91
RV	Right Ventricle. 54, 56
SIMD	Single Instruction Multiple Data. 68, 69
SSE	Sum of Squared Errors. 72
TV	Total Variation. 42, 45, 93, 95, 96, 98, 104, 106, 137, 138

Foreword

The medical discipline of cardiac electrophysiological modeling receives a lot of attention and funding, and will continue to do so. As an interdisciplinary field, settled somewhere between mathematical, physiological, chemical, and electrical sciences, it bears a lot of potential for improvements through cooperations in order to strive to diminish ailments that have only come to humankind's attention in the past few decades. These revelations have been fueled by the rise of computer aided diagnostic medicine, more particularly the field of cardiology for maladies regarding the heart. Such treatments and diagnostic devices are being constantly improved and highly benefit from the advances in computational sciences, bringing them closer to be usable in clinics around the world.

One of the main drawbacks of such interdisciplinary fields is that it requires at least a basic understand in possibly several fields. In order to better understand the underlying fundamentals, a few later discussed pioneering translational works already shed light at multiple aspects in great detail. This thesis starts with introductory knowledge on both ends (mathematical and physiological) that is aimed to provide a foundational understanding of the later used concepts. The ultimate goal of identifying electrophysiological parameters will then be achieved by proposed methods based on these foundations. In these later chapters, you will find the collected knowledge and research output of my colleagues and me over the past four and a half years, along with several conducted experiments that showcase their capabilities. I hope this thesis can, similarly to other works, act both as a translational study to showcase the origin of physiological models, their mathematical (approximative) models, and finally how we can use all of this knowledge and apply it to the inverse problem: Tuning selected parameters of physiological plausible models to best describe measured cardiac electrical activations. While the work heavily leans on mathematical concepts, it is also aimed at researchers and interested individuals of all disciplines, trying to get a head start into the necessary concepts and understandings to start modeling in cardiac electrophysiology.

I hope you enjoy reading the present work and can find something of interest to you.

1 Introduction & Motivation

Cardiac related diseases and genetic disorders are among the leading causes of death in modern developed countries [106]. Yearly, approximately 17.9 million people worldwide die as a consequence of cardiovascular diseases. This necessitates many studies to further understand the inner processes of the heart. While it is common to diagnose heart-related arrhythmias (such as Atrial Fibrillation (AF)) by finding certain shapes and patterns in the Electrocardiogram (ECG) [38], further insights into the inner workings of the healthy and pathological heart would benefit endeavors in cardiac related disease treatment and surgical interventions, as well as drug-development. Research institutes across the world have already picked up on the importance for personalization of cardiac models. However, the approaches vary greatly over applications, practical usages and theoretical foundations. An important foundation for proper personalization of cardiac models already lies in the underlying assumptions and physics related insights of how electrical activation spreads throughout the heart. This topic is now commonly referred to as "Cardiac Electrophysiology". In order to properly model a heartbeat, it is paramount to understand its origin and defining physiology. Pioneering works [51, 78], suggest how the underlying physics can be digitally reproduced using mathematical models. We will shortly summarize some of the touched concepts in Chapter 3 to show how the emergent electrical propagation on a macroscopic level is a consequence of many microscopic interactions. An important property of the emergent behavior is the electrical anisotropy of the tissue as a consequence of the fiber alignment inside the heart [32], which has a prominent role both on its mechanical and electrical function. From an electrophysiological viewpoint, electrical conduction along the fiber direction is generally higher than along cross-fiber direction [29], but also defines the direction in which the maximum mechanical force will be exerted.

With these mathematical models, we can try to digitally recreate encountered physical phenomena (in particular the mentioned electrical activation of the heart in this work) and compare it with data observed and measured in the real world. A modern term commonly used in this regard is "digital twinning", i.e. creating a digital replica called "digital twin", that tries to emulate real world processes [35], such as cardiac electrophysiology. While the forward problem (simulating electrical activation for given electrical parameters) has seen some serious advances [87, 88, 95], fully automated functional twinning — through inverse procedures using sparse measurements — is still in its infancy. The desired computational models/twins of cardiac electrophysiology show high promise not only as clinical research tools, but also as a modality for assisting diagnoses [3] and treatment planning [98]. The full clinical utility could be achieved by personalization of the models to a given patient's electrophysiology based on non- or minimally invasive acquired routine clinical data, such as the ECG. In this context, personalization procedures are also required to be sufficiently fast and automated to be compatible with clinical workflows and time scales.

Meeting such demanding time constraints can be challenging and requires us to strike a balance between high-fidelity modeling of physical processes and desired performance. The aforementioned mathematical models, necessary to approximate electrophysiological behavior in the heart, can be efficiently described in one or multiple Partial Differential Equation (PDE) formulations. Such PDE models can benefit from modern General Purpose Graphics Processing Unit (GPGPU) architecture to gain unprecedented speedups on single machine architectures [89, 122].

The main equation considered throughout this work is the anisotropic eikonal equation, which has been shown to efficiently model cardiac electrical propagation at lower resolutions than its alternatives, the mono- and bidomain equations [33]. An additional benefit of the eikonal equation is that we need not solve a parabolic PDE (i.e. in time and space), but only a first order, albeit non-linear PDE. The non-linearity of the equation will require several mathematical considerations, all of which are discussed in Chapter 2.

There are however several complicating factors that prevent us from using a single, simple anisotropic eikonal model to simulate the electrical behavior throughout the whole heart: From a macroscopic viewpoint, the heart is activated from the sinoatrial node, located in the Right Atrium (RA) and first propagated through Left Atrium (LA) and RA, before being slightly delayed at the atrioventricular node. The delayed ventricular activation then starts from the atrioventricular node rapidly through the His-Purkinje system (HPS) system, where the right and left bundle activate their respective ventricles. The tissue itself propagates the electrical activation to the still unexcited tissue, but at a much slower pace. Figure 1.1 depicts a schematic representation how this propagation looks like on a cross-section of the heart.

This complicated propagation is usually broken down into several stages, some of which are coupled between different systems (e.g. ventricular activation and the HPS propagation). Similarly, we also consider here the different problems in separate methods that model the different parts of the heart: Section 4.6 focuses on the atrial activation, while in Sections 4.5 and 4.7, we consider the ventricular activation.

All of these considerations help us in posing the forward problem in an efficient model that is still true to nature. As indicated by the title of this thesis, this is only an objective to make ends meet from a computational point of view for the personalization of cardiac modeling. Having efficient models that can be evaluated in the timeframe of minutes or less are important when considering the inverse problem of digital twinning. Inverse PDE problems are often posed as minimization problems given noisy data, which can not be directly incorporated into Dirichlet boundary conditions in the original PDE formulation as they inherently may violate the given PDE. In these cases, we need to evaluate the model

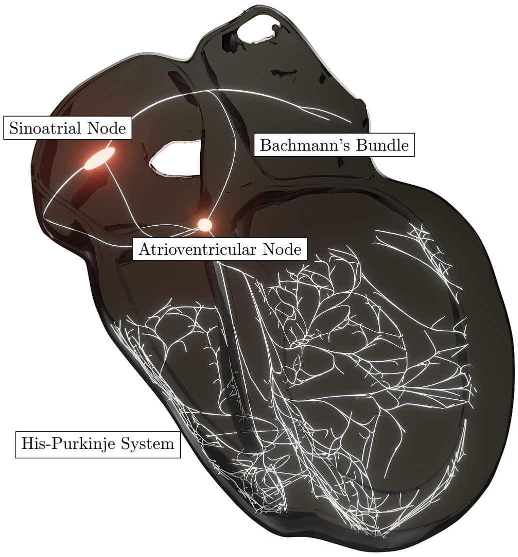


Figure 1.1: Schematic representation of the electrical propagation throughout the heart. The sinoatrial node marks the point of (healthy) electrical initiation, while the atrioventricular node is the pathway to the ventricular activation. In the ventricles, electrical propagation is mediated through the HPS.

multiple times to iteratively match the model's output to the mentioned data and measurements in e.g. a least squares sense. In addition this topic is further complicated by the fact that already the forward problem of solving a PDE often constitutes an optimization problem.

When approaching the topic of optimization from a practical point of view, i.e. computing the inverse problem for already given PDE solvers, the difficulty of implementing optimization on top of the existing framework is one of the main factors. This usually favors derivative-free optimization at a first glance (e.g. evolutionary algorithms), as such black-box optimization procedures require little to no change to existing software implementations to be used in optimization. However, they tend to not scale very well with the number of dimensions/parameters. The problem of exponential growth of the parameter space is commonly referred to as the *curse of dimensionality* [14] and while it not only applies derivative-free optimization, it tends to have a major impact on these optimization methods [27]. In contrast, first order based methods compute not only the solution of a method, but also its (sub)gradient w.r.t. a chosen loss function. The gradient holds a lot of information on the loss function and can be used to represent its linearized approximation around the current point. It has been shown that gradient based optimization schemes can often achieve quadratic convergence for convex problems [13, p. 294], whereas evolutionary algorithms include a lot of randomness that makes the analysis of their convergence rate difficult. Gradient based optimization algorithms are therefore arguably among the most famous methods for Machine Learning (ML)-related optimization [79]. Throughout this work, we will employ many first order gradient methods that performed well in all experiments and were able to minimize a given cost functional (at least locally) to solve the inverse problem of cardiac electrophysiology.

We consider multiple different methodologies in this thesis, all rooted in similar physiological foundations, to model the problem of cardiac electrophysiology. The methodologies between the different works vary in several aspects:

- Different domains are considered (e.g. manifolds, Section 4.6.2)
- The formulation of the problem in terms of the anisotropic eikonal equation, or alternatively its Hamilton–Jacobi formulation (Section 4.7)
- The considered forward model can also rely on ML methods, rather than the traditional Finite Element Method (FEM) formulation (Section 4.6.3)

All in all, the collected works in this thesis should give a broad overview, how the problem of cardiac electrophysiology can be, on the one hand, posed as an effective, quickly solvable model, and, on the other hand, how we can use the model assumption to compute the gradient w.r.t. a meaningful loss function and subsequently optimize this loss to achieve a personalized model rooted in real-world physics.

1.1 Structure of the Work

The remainder of the work is organized up as follows: We start in Chapter 2 by defining the most important used mathematical functions, spaces and their properties, used throughout this work. The content of this chapter is focused around building a foundational understanding of all mathematical models involved in this thesis.

Chapter 3 will deal with the physiological phenomena that make up a heart beat: We describe the chemistry and underlying physiology, first at the level of cells and use the insights of this research that allows to abstract this model, first into a series of discrete circuits and later into PDEs. It will explain the chain of how the electrical impulse is generated from chemical processes at cell level and how these impulses are measured at ECG level. All of these descriptions are accompanied by mathematical formulations that we will later rely on.

In Chapter 4, we review already existing works also targeting the inverse problems in cardiac electrophysiology. We then present a collection of our published methods and discuss model assumptions together with the results on different experiments either from generated models (in-silico), or measured in live patients (in-vivo).

Chapter 5 contains some final insights that are the results of the accumulated published works. These include how the presented works might be further built upon, but also what the results may mean for the future of personalized healthcare, including remaining challenges.

1.2 Discussed Publications

This a list of publications which include me as an author, all of which are discussed or at least touched in this thesis. The full list of publications is available in Chapter B.

> [59] T. Grandits, K. Gillette, A. Neic, J. Bayer, E. Vigmond, T. Pock, and G. Plank. An inverse Eikonal method for identifying ventricular activation sequences from epicardial activation maps. *Journal of Computational Physics*, 419:109700, October 2020, `http://www.sciencedirect.com/science/article/pii/S0021999120304745`

This first work investigated how we can directly use a solver for the anisotropic eikonal equation and implement custom backpropagation to fit the conduction velocity tensors and onset timing to an observed activation on a subset of the domain. We then tested this method called FIMIN for the problem of transmural propagation from epicardial measurements. Discussed in Section 4.5.

> [61] T. Grandits, S. Pezzuto, J. M. Lubrecht, T. Pock, G. Plank, and R. Krause. PIEMAP: Personalized Inverse Eikonal Model from Cardiac Electro-Anatomical Maps. In E. Puyol Anton, M. Pop, M. Sermesant, V. Campello, A. Lalande, K. Lekadir, A. Suinesiaputra, O. Camara, and A. Young, editors, *Statistical Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges*, Lecture Notes in Computer Science, pages 76–86, Cham, 2021. Springer International Publishing

> [83] J. M. Lubrecht, T. Grandits, A. Gharaviri, U. Schotten, T. Pock, G. Plank, R. Krause, A. Auricchio, G. Conte, and S. Pezzuto. Automatic reconstruction of the left atrium activation from sparse intracardiac contact recordings by inverse estimate of fibre structure and anisotropic conduction in a patient-specific model. *EP Europace*, 23(Supplement_1):i63–i70, March 2021, `https://doi.org/10.1093/europace/euaa392`

Modern clinical hardware can generate Electro-Anatomical Maps (EAMs) on the fly during clinical interventions. In such a scenario, a surface model with sparse measurements is built that could be used for an inverse algorithm, such as the previously mentioned work [59]. However, the setup is different, as the generated surface model can be represented as a Riemannian manifold on which we are trying to find both the Local Activation Times (LATs), conduction velocities and fibers. In [61], we defined an inverse model called PIEMAP for EAMs, that estimated all of the mentioned quantities at once. We further tested this method then on several patient-recorded EAMs, as reported in [83]. Discussed in Section 4.6.2.

---

[60] T. Grandits, S. Pezzuto, F. S. Costabal, P. Perdikaris, T. Pock, G. Plank, and R. Krause. Learning Atrial Fiber Orientations and Conductivity Tensors from Intracardiac Maps Using Physics-Informed Neural Networks. In D. B. Ennis, L. E. Perotti, and V. Y. Wang, editors, *Functional Imaging and Modeling of the Heart*, Lecture Notes in Computer Science, pages 650–658, Cham, 2021. Springer International Publishing

---

A very recent approach to both forward and inverse PDE modeling involves Physics Informed Neural Networks (PINNs), which try to estimate the solution of PDEs by learning an Artificial Neural Network (ANN) mapping from coordinates to the sought quantity. We build on this technology by applying the inverse methods of [61] and comparing their quality. Discussed in Section 4.6.3

---

[58] T. Grandits, A. Effland, T. Pock, R. Krause, G. Plank, and S. Pezzuto. GEASI: Geodesic-based Earliest Activation Sites Identification in cardiac models. *International Journal for Numerical Methods in Biomedical Engineering (accepted for publication)*, February 2021, `http://arxiv.org/abs/2102.09962`. arXiv: 2102.09962

The Earliest Activation Sites (EASs) of the eikonal equation (points of electrical initiation inside the heart) are not trivial to find using classical optimization schemes. In this work, we investigate how we can reformulate the problem to arrive at a formulation which is continuous w.r.t. the EASs. The resulting method, called GEASI, is discussed in Section 4.7.

2 Mathematical Preliminaries

> *[...], but in fact mathematicians are like theologians: we regard existence as the prime attribute of what we study. But unlike theologians, we need not always rely upon faith alone.* (Lawrence C. Evans [49])

Contents

2.1 Notation . . . 9
2.2 Functional Analysis . . . 10
2.3 Matrices & Metrics . . . 11
2.4 Continuous Functions . . . 13
2.5 Non-continuous Functions . . . 14
2.6 Ordinary Differential Equations . . . 16
2.7 Differential Geometry . . . 18
2.8 Partial Differential Equations (PDEs) . . . 23
2.9 Finite Element Method (FEM) . . . 29
2.10 Optimization . . . 34

This chapter introduces all of the necessary mathematical preliminaries, needed throughout the whole thesis. As the different works rely on different fields of mathematics, this chapter encompasses the combined needed theories, still irrespective of the underlying physics which will be discussed in Chapter 3.

2.1 Notation

The discrete set of integers betweens 1 and N is given by $[\![N]\!] := \{1, \dots, N\}$.

The notation $|\cdot|$ has multiple meanings depending on the context:

- $|x|, x \in \mathbb{R}$ denotes the absolute value of x
- For a finite countable set, e.g. $[\![N]\!]$, $|N|$ defines the cardinality of said set
- Finally, for continuous domains $\Omega \subset \mathbb{R}^n$, $|\Omega| = \int_\Omega 1$ denotes the size of the domain through means of the Lebesgue measure.

Similarly, δ holds multiple meanings in this thesis according to the context:

- $\delta_C(\mathbf{x})$ for a (convex) set C denotes the indicator function, i.e. $\delta_C(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in C \\ 1 & \text{else} \end{cases}$
- $\delta_{i,j}$ for two integers, $i, j \in \mathbb{N}$ denotes the Kronecker-delta, i.e. $\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$
- In the context of geodesics, for two points on a domain $\mathbf{x}, \mathbf{y} \in \Omega \subset \mathbb{R}^n$, $\delta(\mathbf{x}, \mathbf{y})$ is used to describe the geodesic distance between $\mathbf{x}$ and $\mathbf{y}$.

The notation $\mathbf{x}_i$ refers to the i-th scalar entry of a vector $\mathbf{x} \in \mathbb{R}^n$, whereas $A_{i,j}$ for $1 \leq i \leq n$, $1 \leq j \leq m$ and a matrix $A \in \mathbb{R}^{n \times m}$ denotes the scalar entry at the i-th row and j-th column.

The partial derivatives of a function $u : C^1(\mathbb{R}^n, \mathbb{R})$ are sometimes denoted as $u_{x_1} := \frac{\partial u(\mathbf{x})}{\partial x_1}$. A dot over a function (e.g. $\dot{u}$) signals the derivative w.r.t. a time variable t, i.e. $\dot{u} := \frac{\partial u(t)}{\partial t} = u_t$, and similarly for vectors $\dot{\mathbf{u}} = (\frac{\partial \mathbf{u}_1}{\partial t}, \dots, \frac{\partial \mathbf{u}_n}{\partial t})^\top$. Furthermore, the most commonly used partial derivatives of order 1 and 2, used throughout this work, are the gradient $\nabla u(\mathbf{x}) = \left(\frac{\partial u(\mathbf{x})}{\partial x_1}, \dots \frac{\partial u(\mathbf{x})}{\partial x_n}\right)^\top$ and the Hessian matrix

$$\nabla^2 u(\mathbf{x}) := \begin{pmatrix} u_{x_1,x_1} & \dots & u_{x_1,x_n} \\ & \ddots & \\ u_{x_n,x_1} & \dots & u_{x_n,x_n} \end{pmatrix},$$

respectively. This definition are special cases of the multi-index definition ∇^α of partial derivatives found in [49, p. 701]: $\alpha = (\alpha_1, \dots, \alpha_n)$, $|\alpha| = \sum_i \alpha_i$, for each α_i being a nonnegative integer, such that

$$\nabla^\alpha f(\mathbf{x}) := \frac{\partial^{|\alpha|} f(\mathbf{x})}{\partial \mathbf{x}_1^{\alpha_1} \dots \partial \mathbf{x}_n^{\alpha_n}}.$$

Note that the Hessian is not to be confused with the Laplacian, i.e. $\nabla^2 u(\mathbf{x}) \neq \Delta u(\mathbf{x}) = \nabla \cdot \nabla u(\mathbf{x})$. The Jacobi matrix of a mapping $u : \mathbb{R}^n \supset U \to \mathbb{R}^m$ w.r.t. parameters $\mathbf{x}$ is given by $(J_{\mathbf{x},u})_{i,j} = \frac{\partial u_i(\mathbf{x})}{\partial x_j}$. In the case of real values functions, i.e. $u : U \to \mathbb{R}$, the Jacobi matrix is equal to the transposed gradient $J_{\mathbf{x},u} = \nabla_{\mathbf{x}} u^\top$.

Additionally, for two functions f and g, we say that $f = o(g)$ as $x \to x_0$, iff $\lim_{x \to x_0} \frac{|f(x)|}{|g(x)|} = 0$. Similarly, $f(x) = \mathcal{O}(g(x))$ if $\limsup_{x \to x_0} \frac{|f(x)|}{g(x)} < \infty$.

2.2 Functional Analysis

Throughout this thesis, we usually assume the underlying space of functions to be a real-numbered inner product space, with the following properties [34]:

Definition 2.1 (Inner Product Space)**.** *Let $\mathbb{V}$ be a vector space over $\mathbb{R}$. Then the inner product $v : \mathbb{V} \times \mathbb{V} \to \mathbb{R}$ is a function defined such that for every $\alpha, \beta \in \mathbb{R}$ and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{V}$,* *the following conditions hold:*

(a) $v(\alpha\mathbf{x}+\beta\mathbf{y},\mathbf{z}) = \alpha v(\mathbf{x},\mathbf{z})+\beta v(\mathbf{y},\mathbf{z})$ *(which implies* $v(\mathbf{0},\mathbf{y}) = v(0\mathbf{x},\mathbf{y}) = 0v(\mathbf{x},\mathbf{y}) = 0$*)*
(b) $v(\mathbf{x},\mathbf{x}) \geq 0$
(c) $v(\mathbf{x},\mathbf{y}) = v(\mathbf{y},\mathbf{x})$
(d) $v(\mathbf{x},\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$

The inner product in this thesis will always be denoted as one of the following $v(\mathbf{x},\mathbf{y}) = \langle\mathbf{x},\mathbf{y}\rangle = \mathbf{x}^\top\mathbf{y} = \mathbf{x}\cdot\mathbf{y}$. From this definition of the inner product, we can immediately conclude the definition of (induced) norms $||\mathbf{x}|| = \sqrt{\langle\mathbf{x},\mathbf{x}\rangle}$ where for $\mathbf{x},\mathbf{y} \in \mathbb{V}$ the following holds:

(a) $||\mathbf{x}+\mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}||$ for $\mathbf{x},\mathbf{y} \in \mathbb{V}$ (Triangle inequality)
(b) $||\alpha\mathbf{x}|| = |\alpha|\,||\mathbf{x}||$ for $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{V}$ (Homogeneity)
(c) $||\mathbf{x}|| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$ (Definiteness)

We will most commonly use the p-norms throughout this work ($\mathbf{x} \in \mathbb{R}^n$), defined as

$$||\mathbf{x}||_p \coloneqq \left(|x_i|^p\right)^{1/p}, \tag{2.1}$$

with the particularly important cases

- $||\mathbf{x}||_1 = \sum_{i=1}^n |\mathbf{x}_i|$
- $||\mathbf{x}||_\infty = \max_{i=1,\dots,n} |\mathbf{x}_i|$.

Definition 2.2 (Dual Norms)**.** *Consider a norm* $||\cdot||$ *on* $\mathbb{R}^n$*. The associated dual norm* $||\cdot||_*$ *is given by*

$$||\mathbf{y}||_* \coloneqq \sup_{\mathbf{x}} \left\{\langle\mathbf{x},\mathbf{y}\rangle \mid ||\mathbf{x}|| \leq 1\right\}. \tag{2.2}$$

Note that for $p \geq 1$, the dual norm of $||\cdot||_p$ is given by $||\cdot||_{p,*} = ||\cdot||_q$ for $\frac{1}{p} + \frac{1}{q} = 1$ [13].

An inner product space is said to be a Hilbert space, if it is complete, i.e. every Cauchy sequence in $\mathbb{V}$ converges to an element of $\mathbb{V}$ [125]. We define the L^2 space as the space of square integrable functions

$$||f|| = \left(\int |f|^2 \,\mathrm{d}\mathbf{x}\right)^{1/2} < \infty \tag{2.3}$$

2.3 Matrices & Metrics

Consider the real square matrix $A \in \mathbb{R}^{n\times n}$. $\operatorname{rank}(A)$ is equal to its number of linearly independent rows. The vectors $\mathbf{v}_i \in \mathbb{C}^n \setminus \{\mathbf{0}\}$ are called eigenvectors, if there exists $\sigma_i \in \mathbb{C}$, such that:

$$A\mathbf{v}_i = \sigma_i\mathbf{v}_i \tag{2.4}$$

In this case, σ_i is called the eigenvalue associated to the corresponding eigenvector $\mathbf{v}_i$. For $A \in \mathbb{R}^{n \times n}$ there exist exactly n eigenvalues and eigenvectors. Note that if A is symmetric, we can additionally prove orthogonality of the eigenvectors ($\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$) and that all eigenvalues are real for matrices of full rank ($\operatorname{rank}(A) = n$) with n distinct eigenvalues, i.e. $A^\top = A \Rightarrow \sigma_i \in \mathbb{R}$. Consider the two distinct eigenvectors $\mathbf{v}_i$ and $\mathbf{v}_j$ of A with the associated eigenvalues of σ_i and σ_j respectively:

$$\begin{aligned}
\langle A\mathbf{v}_i, \mathbf{v}_j \rangle &= \left\langle \mathbf{v}_i, A^\top \mathbf{v}_j \right\rangle = \langle \mathbf{v}_i, A\mathbf{v}_j \rangle \\
\sigma_i \langle \mathbf{v}_i, \mathbf{v}_j \rangle &= \sigma_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \\
(\sigma_i - \sigma_j) \langle \mathbf{v}_i, \mathbf{v}_j \rangle &= 0
\end{aligned}$$

which, if we assume $\sigma_i \neq \sigma_j$, is only the case for $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$.

$$\begin{aligned}
A\mathbf{v}_i &= \sigma_i \mathbf{v}_i \\
\langle A\mathbf{v}_i, A\mathbf{v}_i \rangle &= \mathbf{v}_i^* A^\top A \mathbf{v}_i = \sigma_i^2 \left|\left|\mathbf{v}_i\right|\right|^2 \\
\sigma_i^2 &= \frac{\langle A\mathbf{v}_i, A\mathbf{v}_i \rangle}{\left|\left|\mathbf{v}_i\right|\right|^2},
\end{aligned}$$

which is a real positive number as the norm on complex numbers $\mathbf{v}_i \in \mathbb{C}^n$ is given by $||\mathbf{v}_i|| = \sqrt{\mathbf{v}_i^* \mathbf{v}_i} = \sqrt{\sum_j \left|(\mathbf{v}_i)_j\right|^2}$. If A is symmetric therefore, it can then be expressed in terms of its eigendecomposition as:

$$A = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}}_{U} \underbrace{\operatorname{diag}(\sigma_1, \dots, \sigma_n)}_{\Sigma} \underbrace{\begin{bmatrix} — & \mathbf{v}_1 & — \\ — & \mathbf{v}_2 & — \\ & \vdots & \\ — & \mathbf{v}_n & — \end{bmatrix}}_{U^\top} \tag{2.5}$$

A special case of square matrices, heavily used later on, are symmetric positive definite matrices $A \in S^n_{++} \subset \mathbb{R}^{n \times n}$, defined as:

$$S^n_{++} = \left\{ A \,|\, A = A^\top \wedge \mathbf{x}^\top A \mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\} \right\}, \tag{2.6}$$

and similarly, the set of symmetric positive semi-definite matrices

$$\bar{S}^n_{++} = \left\{ A \,|\, A = A^\top \wedge \mathbf{x}^\top A \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\} \right\}. \tag{2.7}$$

We can express (2.6) in terms of the eigenvalues shown in (2.5):

$$\begin{aligned}\mathbf{x}^\top A\mathbf{x} &= \mathbf{x}^\top U\Sigma \underbrace{U^\top \mathbf{x}}_{\tilde{\mathbf{x}}} \\ &= \tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}} = \sum_{i=1}^{n} \sigma_i \tilde{x}_i^2 > 0,\end{aligned} \tag{2.8}$$

since Σ is a diagonal matrix. From (2.8) we can see that $A \in S_{++}^n$ iff all eigenvalues of A are strictly positive, i.e. Σ is a strictly positive diagonal matrix.

This special class of matrices can be used to induce a new metric in a vector space:

$$\begin{aligned}\langle \mathbf{x}, \mathbf{x}\rangle_A &= \mathbf{x}^\top A\mathbf{x} \geq 0 \\ ||\mathbf{x}||_A &= \sqrt{\langle \mathbf{x}, \mathbf{x}\rangle_A} \geq 0\end{aligned} \tag{2.9}$$

which complies with all introduced conditions for a vector space in Section 2.2. A common representation, also found later in this work, is the ellipsoid visualizing the level line $||\mathbf{x}||_D = 1$, like can be seen in Figure 2.1. This ellipsoid can also be interpreted as being spanned by its eigenvalues in the directions of the eigenvectors. Note that the dual norm $||\cdot||_{A,*}$ is the norm in the inverse metric $||\cdot||_{A^{-1}}$.

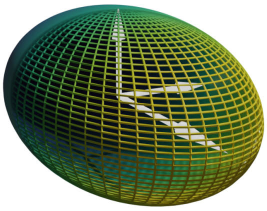


Figure 2.1: Common representation of a tensor $D \in S_{++}^3$. The ellipsoid represents the contour line that satisfies $||\mathbf{x}||_D = 1$, while the arrows show the eigenvectors scaled with their respective eigenvalue.

2.4 Continuous Functions

Before we consider non-continuous functions and their properties, we shortly review the definition of continuous and continuously differentiable functions. Many of the definitions are given for functions $f : \mathbb{R}^n \to \mathbb{R}$, as it is the most considered class of functions throughout this thesis. The class of continuous functions is given through the definition

Definition 2.3 (Continuous Functions [36]). *We call a function $f : U \to \mathbb{R}$ continuous ($f \in C^0$) if the limit*

$$\lim_{\mathbf{x}\to\xi} f(\mathbf{x})$$

exists and coincides with $f(\xi)$ for all points $\xi \in U$.

Using Definition 2.3, we can define a function to be continuously differentiable $f \in C^1$, if all of its partial derivatives $\frac{\partial f}{\partial x_i}$ exist and are continuous. Similarly, we can define k-times continuously differentiable functions $f \in C^k$ as functions for which all partial derivatives of up to order k exist and are continuous. In general, it holds that $C^k \subset C^n$ for $n > k$. Throughout this work, $C^k(R^n, R)$ denotes the space of k times continuously differentiable functions $f : \mathbb{R}^n \to \mathbb{R}$.

2.5 Non-continuous Functions

Continuous and continuously differentiable functions are very important to consider, but later introduced problems, such as the eikonal equation, require weaker definitions of continuity. We start by defining semi-continuous functions [43] using the upper and lower semi-continuous envelopes:

$$\begin{cases} u^*(\mathbf{x}) = \limsup_{r\to 0} \{u(\mathbf{y}) : \mathbf{y} \in U,\ ||\mathbf{y}-\mathbf{x}|| \leq r\} \\ u_*(\mathbf{x}) = \liminf_{r\to 0} \{u(\mathbf{y}) : \mathbf{y} \in U,\ ||\mathbf{y}-\mathbf{x}|| \leq r\} \end{cases} \tag{2.10}$$

Definition 2.4. *A function u is called upper semi-continuous if $u = u^*$ and lower semi-continuous if $u = u_*$. If a function is simultaneously both upper- and lower semi-continuous, it is also continuous and vice versa.*

This definition of semi-continuity can be used to define sub- and superdifferentials:

Definition 2.5. *The set of super- and subdifferentials ($\nabla^+ u$ and $\nabla^- u$ respectively) at a point $\mathbf{x} \in U$ are defined as*

$$\begin{cases} \nabla^+ u(\mathbf{x}) := \left\{\mathbf{p} \in \mathbb{R}^n :\ \limsup_{\mathbf{y}\to\mathbf{x}} \frac{u(\mathbf{y})-u(\mathbf{x})-\langle \mathbf{p},\mathbf{y}-\mathbf{x}\rangle}{||\mathbf{y}-\mathbf{x}||} \leq 0\right\} \\ \nabla^- u(\mathbf{x}) := \left\{\mathbf{p} \in \mathbb{R}^n :\ \liminf_{\mathbf{y}\to\mathbf{x}} \frac{u(\mathbf{y})-u(\mathbf{x})-\langle \mathbf{p},\mathbf{y}-\mathbf{x}\rangle}{||\mathbf{y}-\mathbf{x}||} \geq 0\right\} \end{cases} \tag{2.11}$$

In more colloquial terms, we say that the superdifferentials $\nabla^+ u(\mathbf{x})$ is the set of supporting hyperplanes above a function at $\mathbf{x}$. $\nabla^- u(\mathbf{x})$ similarly is the set of supporting hyperplanes below the function at $\mathbf{x}$. In regions of continuous differentiability, it holds that $\nabla^+ u(\mathbf{x}) = \nabla^- u(\mathbf{x}) = \{\nabla u(\mathbf{x})\}$, i.e. both super- and subdifferential are well defined and equal to the unique gradient.

Remark 2.1. *Definition 2.5 defines super- and subdifferentials only locally around points as the supporting hyperplanes above and below the functions respectively. Later formulations in Section 2.10 require the supporting hyperplane to bound the function from above/below on the whole domain $\mathbf{x} \in U$ (see [102]), which we will denote with ∂. We* *only consider the locally supporting hyperplanes for now, as we later need them to define viscosity solutions in Section 2.8.1.*

Example 2.1. [The $||\cdot||_1$ Cone] Consider the l_1-norm cone in 2D ($\mathbf{x} \in \mathbb{R}^2$), given by $f(\mathbf{x}) = ||\mathbf{x}||_1 = |\mathbf{x}_1| + |\mathbf{x}_2|$. f is continuously differentiable on the whole domain except for $\mathbf{x}_1 = 0 \vee \mathbf{x}_2 = 0$ (i.e. almost everywhere). For the absolute function $|x|$ we can define the subdifferential at $x = 0$ according to (2.11) as

$$\nabla^- |x| = \operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ [-1, 1] & \text{else} \end{cases}$$

This allows us then to define the subdifferential $\nabla^- f$ as:

$$\nabla^- f(\mathbf{x}) = \left\{ \begin{pmatrix} p_1 \in \operatorname{sgn}(x_1) \\ p_2 \in \operatorname{sgn}(x_2) \end{pmatrix} \right\}$$

♦

For functions that are not continuously differentiable everywhere, we can introduce a weaker definition of continuity: Lipschitz continuous functions are themselves continuous, but not continuously differentiable [19]. The space of Lipschitz continuous functions is a special case of the more general Hölder spaces:

Definition 2.6. *$C^{k,\alpha}$ is the space of k times differentiable functions (called Hölder space) for which the norm*

$$||f||_{C^{k,\alpha}} = ||f||_{C^k} + \max_{|\beta|=k} \left| \nabla^\beta f \right|_{C^{0,\alpha}}$$

is finite. $||f||_{C^{0,\alpha}}$ is defined as

$$||f||_{C^{0,\alpha}} = \sup_{\mathbf{x} \neq \mathbf{y} \in \Omega^2} \frac{|f(\mathbf{x}) - f(\mathbf{y})|}{||\mathbf{x} - \mathbf{y}||^\alpha}.$$

In particular, the space $C^{0,1}$ fulfilling

$$||f(\mathbf{x}) - f(\mathbf{y})|| \leq \bar{L} \, ||\mathbf{x} - \mathbf{y}||$$

is called the space of Lipschitz continuous functions. We refer to the constant $\bar{L}$ as the Lipschitz constant of the function f.

In this context, it is often helpful to extend our concept of classical derivatives with weak derivatives.

Definition 2.7 (Weak Derivatives [1]). *Let u be in the space of locally integrable functions $L^1_{loc}(\Omega)$ for $\Omega \subset \mathbb{R}^n$. $\nabla^\alpha u = v_\alpha$ is then the α-th weak derivative of u if it satisfies*

$$\int_\Omega u(\mathbf{x}) \nabla^\alpha \phi(\mathbf{x}) \, \mathrm{d}\mathbf{x} = (-1)^{|\alpha|} \int_\Omega v_\alpha(\mathbf{x}) \phi(\mathbf{x}) \, \mathrm{d}\mathbf{x},$$

for $\phi \in C^\infty(\Omega)$ with compact support in Ω. If such a v_α exists, it is unique up to sets of Lebesgue measure zero.

We continue by also introducing Sobolev norms,

Definition 2.8 (Sobolev Norm [1]). *The Sobolev norm is defined as the functional $||\cdot||_{m,p}$, for m being a positive integer and $1 \leq p < \infty$, as*

$$||f||_{m,p} = \left(\sum_{|\alpha| \leq k} \int_\Omega |\nabla^\alpha f(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} \right)^{1/p} .$$

With the help of Definitions 2.7 and 2.8, we can define the Sobolev space $W^{m,p}$.

Definition 2.9 (Sobolev Space). *We call the space $W^{m,p}$*

$$W^{m,p}(\Omega) = \{ f \in L^p(\Omega) : \nabla^\alpha f \in L^p(\Omega) \text{ for } 0 \leq |\alpha| \leq m \} , \tag{2.12}$$

a Sobolev space where ∇^α is the weak partial derivative according to Definition 2.7.

Throughout this thesis, we will only look at the space $W^{m,2}$. Colloquially, we say f is differentiable almost everywhere, i.e. regions with non-continuous gradients have zero Lebesgue measure. Note that $W^{m,2}$ has been shown to be equal to the Hilbert space H^m [1].

2.6 Ordinary Differential Equations

This work will only shortly focus on Ordinary Differential Equations (ODEs), which is why we only introduce the most important definition and properties. All of the introduced concepts of this section are based on and described in much more detail in [28].

Definition 2.10. *Consider a time-interval $I \subset \mathbb{R}$ and the domain of our time-evolving state variables $\mathbf{x} \in U \subset \mathbb{R}^n$, together with a continuous function $f \in C^0(I \times U, \mathbb{R}^n)$. A system of ODEs can then be written in its general form as*

$$\dot{\mathbf{x}} = f(t, \mathbf{x}) \tag{2.13}$$

where $t \in I$.

As we will later see in Section 3.1, such ODEs are commonly encountered in many biological processes, describing the behavior of such a system over time.

Example 2.2. [Population Growth] Consider the growth of a population $x \in \mathbb{R}$ with a growth factor $r \in \mathbb{R}$, limited by the availability of resources to size $k \in \mathbb{R}$. This problem can be formulated using the model

$$\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t} = rx\left(1 - \frac{x}{k}\right)$$

This equation, though seemingly simple at first, can model many phenomena, such as modern human population growth, or also the uncontrolled spread of infections during a pandemic. We can solve the above ODE using separation of variables to explicitly express $x(t)$:

$$\begin{aligned} r\,\mathrm{d}t &= \left(x\left(1 - \frac{x}{k}\right)\right)^{-1}\mathrm{d}x = \left(x^{-1} + k^{-1}\left(1 - \frac{x}{k}\right)^{-1}\right)\mathrm{d}x \\ rt &= \ln(x) - \ln\left(k\left(1 - \frac{x}{k}\right)\right) + c_1 \\ c\exp(rt) &= \frac{x}{k\left(1 - \frac{x}{k}\right)} \\ x &= \frac{k}{1 + \left(\frac{k - x_0}{x_0}\right)\exp(-rt)} \end{aligned}$$

where $x(0) = x_0$ denotes the initial value. If we choose $r = k = 1$ and $x_0 = 0.5$, we will end up with the well known C^∞ sigmoid function

$$x(t) = \frac{1}{1 + \exp(-t)}, \tag{2.14}$$

visualized in Figure 2.2, which is commonly encountered in many later considered ML applications [60, 100, 109]. ♦

For the assumptions made in Definition 2.10, we can also define existence and uniqueness of a solution that smoothly depends on our choice of initial values in the case $f \in C^{0,1}$. The proof requires us first to define successive approximations to (2.13), given by the recursive formulas

$$\begin{array}{rl} \phi_0(t) & = \mathbf{x}_0 \\ \phi_{k+1}(t) = \mathbf{x}_0 + \int_{t_0}^{t} f(s, \phi_k(s))\,\mathrm{d}s \qquad (k & = 0, 1, 2, \ldots; |t - t_0| \leq \alpha), \end{array}$$

for the initial times and values, t_0 and $\mathbf{x}_0 \in U$ respectively.

Theorem 2.1 (Picard–Lindelöf [31])**.** *Consider a function $f \in C^{0,1}([t_0, t] \times U, \mathbb{R}^n)$, used in the initial value problem (2.13), which is Lipschitz in the second argument. Then the*

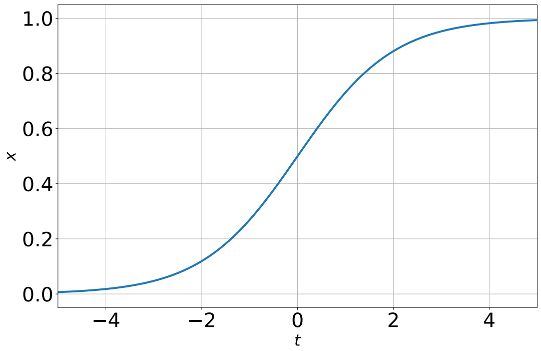


Figure 2.2: The sigmoid function, given by (2.14) is commonly encountered in many ML applications.

successive approximations ϕ_k exist on $|t - t_0| \leq \alpha$ as continuous functions and converge uniformly on this interval to the unique solution ϕ of (2.13) such that $\phi(t_0) = \mathbf{x}_0$.

In many practical applications we rely on numerical solutions of (2.13), with an arguably prominent example being the explicit forward explicit Euler scheme given by [20]:

$$\begin{cases} \mathbf{x}_{i+1} & = \mathbf{x}_i + hf(t_i, \mathbf{x}_i) \\ t_{i+1} & = t_i + h \end{cases} \tag{2.15}$$

for a small enough step size $h > 0$. Note that if we assume $f(t_i, \mathbf{x}_i) = -\nabla_{\mathbf{x}} g(\mathbf{x}_i)$ to be the negative gradient of a function, Equation (2.15) will yield the famous gradient descent scheme, employed in many optimization methods.

The forward Euler scheme is actually a special case of Runge–Kutta methods which can be more generally formulated using Butcher tableaus [20]. In general, Runge–Kutta methods enable the formulation of higher order solution steps that consequently decrease the order of the resulting approximation error. We later use the second order explicit midpoint scheme in Section 4.7, which is given by:

$$\begin{cases} \mathbf{x}_{i+1} & = \mathbf{x}_i + hf(t_i + \frac{h}{2}, \mathbf{x}_i + \frac{h}{2} f(t_i, \mathbf{x}_i)) \\ t_{i+1} & = t_i + h \end{cases} \tag{2.16}$$

2.7 Differential Geometry

Differential geometry (and its extension, Riemannian geometry) is a vast field of research that can not be hoped to be captured in a small section. We will later use concepts of differential geometry in multiple areas of this work, especially in Sections 4.6.2 and 4.7. The short introduction of used concepts are a very compact representation of thorough explanations and definitions found in the works [24, 81, 91], where the definition of all concepts mostly follow [24]. Note that differentiability in this section refers to the class C^∞.

Definition 2.11 (Differentiable Manifold [24]). *A differentiable manifold of dimension n is a set M and a family of injective mappings $\mathbf{x}_\alpha : \mathbb{R}^n \supset U_\alpha \to M$ such that:*

1. $\bigcup_\alpha \mathbf{x}_\alpha(U_\alpha) = M$
2. *For any pair α, β with $\mathbf{x}_\alpha(U_\alpha) \cap \mathbf{x}_\beta(U_\beta) = W$, the sets $\mathbf{x}_\alpha^{-1}(W)$ and $\mathbf{x}_\beta^{-1}(W)$ are open sets in $\mathbb{R}^n$ and the mappings $\mathbf{x}_\beta^{-1} \circ \mathbf{x}_\alpha$ are differentiable*
3. *The family $\{(U_\alpha, \mathbf{x}_\alpha)\}$ is maximal relative to the conditions 1 and 2*

We call the pair $(U_\alpha, \mathbf{x}_\alpha)$ with $\mathbf{p} \in \mathbf{x}_\alpha(U_\alpha)$ a parameterization of M at $\mathbf{p}$. $\mathbf{x}_\alpha$ is sometimes referred as the mapping.

A visual representation of Definition 2.11 can be found in Figure 2.3. An important concept that we will use in many instances is the tangent space $T_\mathbf{p}M$, associated to the manifold M at point $\mathbf{p} \in M$. For this purpose, we start by considering a differentiable curve $\alpha : (-\varepsilon, \varepsilon) \to \mathbb{R}^n$ with $\alpha(0) = \mathbf{p}$, given by

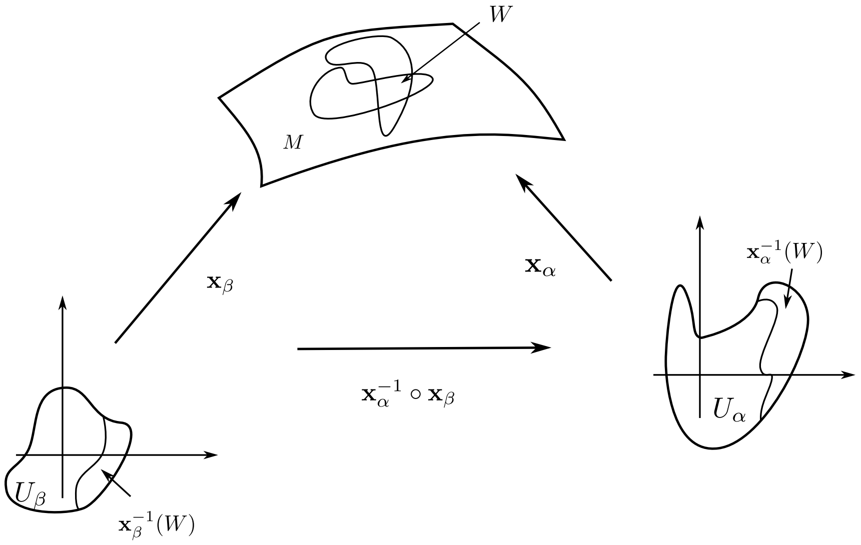

W
M
$\mathbf{x}_\beta$
$\mathbf{x}_\alpha$
$\mathbf{x}_\alpha^{-1}(W)$
$\mathbf{x}_\alpha^{-1} \circ \mathbf{x}_\beta$
U_α
U_β
$\mathbf{x}_\beta^{-1}(W)$

Figure 2.3: Visualization of the definition of a differentiable manifold, in accordance with Definition 2.11. The manifold is formed by the union of multiple differentiable mappings, between which we can change smoothly on W. Similar to [24, p. 3]

$$\alpha(t) = (x_1(t), \dots, x_n(t)),\, t \in (-\varepsilon, \varepsilon),\, (x_1, \dots, x_n) \in \mathbb{R}^n,$$

and define the initial direction as $\mathbf{v} = \frac{\partial \alpha}{\partial t}(0)$. When additionally considering a differentiable function f, defined in a neighborhood of $\mathbf{p}$, we can restrict f to the curve α and express the directional derivative with respect to $\mathbf{v} \in \mathbb{R}^n$ as:

$$\left.\frac{\partial(f \circ \alpha)}{\partial t}\right|_{t=0} = \left.\left\langle \nabla_{\mathbf{x}} f, \frac{\partial \alpha}{\partial t} \right\rangle\right|_{t=0} = \left(\sum_{i=1}^{n} \frac{\partial x_i}{\partial t}(0) \frac{\partial}{\partial x_i} \right) f. \tag{2.17}$$

Note that from the above equation we can see that the directional derivative with respect to $\mathbf{v}$ is an operator on differentiable functions that depends uniquely on $\mathbf{v}$. This is an important property for defining tangent vectors on manifolds:

Definition 2.12 (Tangent Vectors & Spaces)**.** *Consider a differentiable manifold M with a differentiable curve on it $\alpha : (-\varepsilon, \varepsilon) \to M$. For $\alpha(0) = \mathbf{p} \in M$ and $\mathcal{D}$ being the set of functions differentiable functions at $\mathbf{p}$, the tangent vector to the curve α at $t = 0$ is given by the function $\frac{\partial \alpha}{\partial t}(0) : \mathcal{D} \to \mathbb{R}$, for which*

$$\frac{\partial \alpha}{\partial t}(0) f = \left.\frac{\partial \left(f \circ \alpha\right)}{\partial t}\right|_{t=0}, \; f \in \mathcal{D}$$

A tangent vector at $\mathbf{p}$ is therefore a tangent vector at $t = 0$ for some curve $\alpha : (-\varepsilon, \varepsilon) \to M$ for $\alpha(0) = \mathbf{p}$. The set of all tangent vectors of M at $\mathbf{p}$ is called the tangent space, indicated by $T_{\mathbf{p}}M$.

Combining Definition 2.12 and (2.17), we can express the tangent vector at $\mathbf{p}$ as

$$\frac{\partial \alpha}{\partial t}(0) = \sum_{i=1}^{n} \frac{\partial x_i}{\partial t}(0) \left(\frac{\partial}{\partial x_i} \right)_0 . \tag{2.18}$$

We can see from (2.18) that $\left(\frac{\partial}{\partial x_i}\right)_0$ is the tangent vector at $\mathbf{p}$ of the chosen curve. Another important consequence is that $T_{\mathbf{p}}M$ forms a vector space of dimension n with the associated basis

$$\left\{ \left(\frac{\partial}{\partial x_1} \right)_0, \ldots, \left(\frac{\partial}{\partial x_n} \right)_0 \right\}, \tag{2.19}$$

depending on the parameterization $\mathbf{x} : U \to M$, hence the name *tangent space*. We again present a visualization of the concept in Figure 2.4

Definition 2.13 (Riemannian Metrics and Manifolds [24, 91])**.** *A Riemannian metric on a differentiable manifold M (Definition 2.11) is a correspondence which smoothly assigns to each point $\mathbf{p} \in M$ an inner product $\langle \cdot, \cdot \rangle_g$ on the tangent space $T_{\mathbf{p}}M$. A differentiable manifold with a given Riemannian metric is called a Riemannian manifold.*

Note that the Definition 2.13 requires $g \in S^n_{++}$ to be a symmetric positive tensor (Section 2.3), where its coefficients in a local parameterization are given as

$$g_{i,j} = \langle \partial_i, \partial_j \rangle \quad (1 \leq i, j \leq n) \tag{2.20}$$

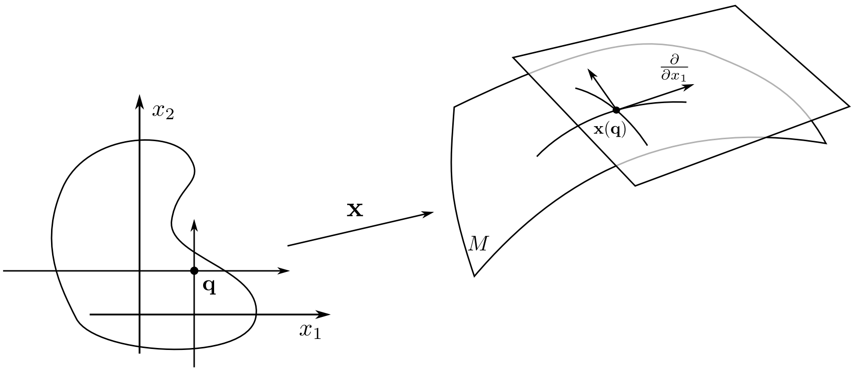


Figure 2.4: Visual representation of a tangent space on M. The two axis aligned curves, intersecting at $\mathbf{q}$ are mapped to M via $\mathbf{x}$. $\frac{\partial}{\partial x_1}$ and $\frac{\partial}{\partial x_2}$ lie in the tangent space at $\mathbf{x}(\mathbf{q})$. Similar to [24, p. 8]

While we already introduced differentiable curves on manifolds that we later need for the theory of geodesics, we will instead rely on the equivalent, but slightly different Hamilton–Jacobi formulation in Section 2.8 to define geodesics. There is however one concept, also later exploited, that relies on differential geometry: The exponential map and its inverse, the logarithmic map.

Definition 2.14 (Exponential & logarithmic map)**.** *Consider a (uniquely defined) geodesic $\gamma_V^{\mathbf{p}} : \mathbb{R} \supset I \to M$ starting at a point $\mathbf{p} \in M$ $(t = 0)$ and initially moving in the direction $V \in T_{\mathbf{p}}M$. In a neighborhood $U \subset T_{\mathbf{p}}M$, the exponential map $\exp_{\mathbf{p}} : V \to M$ given by*

$$\exp_{\mathbf{p}}(V) = \gamma_V^{\mathbf{p}}(1)$$

is well defined. Conversely, its inverse, the logarithmic map, is given by $\log_{\mathbf{p}} : M \to V$.

Computing exponential and logarithmic maps can be a challenging task, as potentially it could require the computation of all geodesics on discrete meshes. In [113] however, a method is given to efficiently compute said logarithmic map on manifolds by means of parallel transportation of vectors using the Levi–Civita connection (not presented here). We will conclude this section with a simple example demonstrating all of the introduced concepts:

Example 2.3. [Hills and Valleys] Consider the 2-dimensional submanifold embedded into $M \subset \mathbb{R}^3$ given by the local parameterization $f : \mathbb{R}^2 \supset \Omega \to M$:

$$f(\xi, \eta) = \Big(\xi, \quad \eta, \quad -\cos 2\xi + 2\sin\eta\Big)^\top$$

for $\Omega = [0, 11]^2$. The inverse mapping is simply given by $f^{-1}(x, y, z) = (x, y)^\top$, so we can see that f is a bijective mapping. The Jacobi matrix for the local parameterization f is given by

$$J_{f,(\xi,\eta)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2\sin 2\xi & 2\cos\eta \end{pmatrix}$$

where $\nabla_\xi f = (1, 0, 2\sin 2\xi)^\top \in T_{f(\cdot,\xi)}M$ and $\nabla_\eta f = (0, 1, 2\cos\eta)^\top \in T_{f(\eta,\cdot)}M$ are two orthogonal tangent space directions. The Riemannian metric for this manifold — inducing the inner-product on the manifold $\langle\cdot,\cdot\rangle_g$ in its parametric ξ, η space — is given by $g(\mathbf{x}) = J_{f,\mathbf{x}}^\top J_{f,\mathbf{x}}$:

$$g(\xi,\eta) = \begin{pmatrix} 1 + 4\sin^2 2\xi & 4\sin 2\xi\cos\eta \\ 4\sin 2\xi\cos\eta & 1 + 4\cos^2\eta \end{pmatrix}$$

The induced inner-product gives us access to measuring lengths on the manifold using the induced norm $||\cdot||_{g(\xi,\eta)}$.

Consider now two curves $\gamma_1 : [0,1] \to \Omega$ and $\gamma_2 : [0,1] \to M$, such that they are assumed "equal" in their opposite mapping, i.e.

$$\begin{aligned} \forall t_1 \in [0,1]\ \exists t_2 \in [0,1] : \quad & f(\gamma_1(t_1)) && = \gamma_2(t_2) \\ \forall t_2 \in [0,1]\ \exists t_1 \in [0,1] : \quad & \gamma_1(t_1) && = f^{-1}(\gamma_2(t_2)) \end{aligned}$$

Through the Riemannian metric, we know that $||\dot\gamma_1(t_1)||_{g(\gamma_1(t_1))} = ||\dot\gamma_2(t_2)||$ for t_1 and t_2 according to the above equation. Assume now additionally that γ_2 is the geodesic path between two points $\mathbf{p}_0 = \gamma_2(0)$ and $\mathbf{p}_1 = \gamma_2(1)$. The geodesic distance $\delta(\mathbf{p}_0, \mathbf{p}_1)$ can then be expressed in the local parameterization as:

$$\delta(\mathbf{p}_0, \mathbf{p}_1) = \int_0^1 ||\dot\gamma_2(t_2)||\,\mathrm{d}t = \int_0^1 ||\dot\gamma_1(t_1)||_{g(\gamma_1(t_1))}\,\mathrm{d}t$$

Inversely, for $\phi(\xi,\eta) \coloneqq \delta(\mathbf{p}_0, f(\xi,\eta))$, the PDE

$$||\nabla\phi||_{g^{-1}} = 1 \quad \text{in } \Omega$$

is satisfied almost everywhere. Figure 2.5 shows the geodesic path in both the 2D domain Ω and on the 3D manifold M.

♦

Remark 2.2. *Example 2.3 already foreshadows the strong links between the anisotropic eikonal equation and Riemannian manifold theory. This fact will later be used in Section 4.7 to introduce an inverse method that uses theorems of Riemannian geometry. Note however that in all methods, there is no specification of a possible underlying manifold in* $\mathbb{R}^{n+1}$ *dimensions and instead we only rely on the knowledge the Riemannian metric tensor* $D(\mathbf{p}) \coloneqq g^{-1}(\mathbf{p})$.

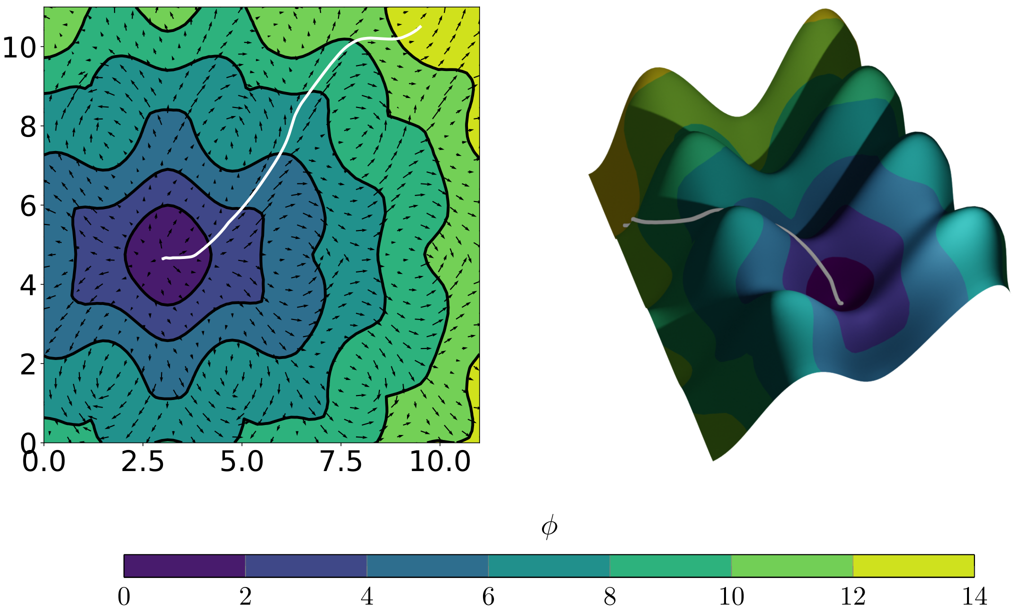


Figure 2.5: Visualization of the manifold example in Example 2.3. The left figure shows how the induced Riemannian metric allows us to view the problem as a simple 2D plane with an induced norm, representing an anisotropic and heterogeneous velocity, whereas the right figure shows the actual underlying Riemannian manifold on which all paths and distances are Euclidean.

2.8 Partial Differential Equations (PDEs)

This section will shortly introduce the notion of PDEs and heavily relies on the definitions and notations from [49] for this purpose. We will however only consider PDEs of order up to 2, as higher order PDEs are not considered in the proposed methods.

Definition 2.15. *Considering an open subset $U \subset \mathbb{R}^n$, the expression*

$$F(\nabla^2 u(\mathbf{x}), \nabla u(\mathbf{x}), u(\mathbf{x}), \mathbf{x}) = 0 \quad \mathbf{x} \in U, \tag{2.21}$$

for a function $F : \mathbb{R}^{n \times n} \times \mathbb{R}^n \times \mathbb{R} \times U \to \mathbb{R}$ is a general formulation of a second-order partial differential equation, where $u : U \to \mathbb{R}$ is the unknown.

Note that first order PDEs are just a special case of this definition for a proper choice of F. A special subclass of such PDEs are called linear in case they can be written as

$$\left\langle \mathbf{a}_2(\mathbf{x}), \nabla^2 u(\mathbf{x}) \mathbf{a}_3(\mathbf{x}) \right\rangle + \left\langle \mathbf{a}_1(\mathbf{x}), \nabla u(\mathbf{x}) \right\rangle + b(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}),$$

for given functions $\mathbf{a}_i : U \to \mathbb{R}^n$ for $i \in \{1, 2, 3\}$ and $b : U \to \mathbb{R}$.

A PDE problem, as per Definition 2.15 is well-posed if the problem has a unique solution that continuously depends on the data given by the boundary conditions. We call solutions to (2.21) that are (for our definition) once or twice continuously differentiable (equal to the order of the problem) *classical solutions* to the problem.

In particular, we will later consider Poisson's equation

$$\begin{cases} \Delta u &= f \quad \text{in } U \\ u &= g \quad \text{on } \partial U \end{cases} \tag{2.22}$$

for $f : U \to \mathbb{R}$, $\Delta u = \sum_{i \in [\![n]\!]} \frac{\partial^2 u}{\partial x_i^2}$ and a given boundary function $g : \partial U \to \mathbb{R}$. We know that in the case $g \in C(\partial U, \mathbb{R})$, there exists at most one solution $u \in C^2(U, \mathbb{R}) \cap C(\overline{U}, \mathbb{R})$ of the boundary value problem [49, Chapter 2, p. 28]. Note that under certain regularity conditions, we could additionally show that $u \in C^\infty$ [49, p. 28].

2.8.1 The Eikonal Equation

The other heavily employed PDE throughout this work is the anisotropic eikonal equation, a first order non-linear PDE given by:

$$\begin{cases} ||\nabla u||_D = \sqrt{\langle D\nabla u, \nabla u\rangle} = 1 & \text{in } U \\ u = g & \text{on } \Gamma \subset \partial U \end{cases} \tag{2.23}$$

for a boundary function $g : \partial U \supset \Gamma \to \mathbb{R}$ and a norm-inducing metric $D \in S^n_{++}$. The solution of (2.23) can be intuitively interpreted as calculating the time it takes to travel through a medium with heterogeneous, anisotropic velocity D to each point in U from the prescribed boundary Γ. Studying and solving (2.23) in a classical sense is in practice not feasible for many examples though:

Example 2.4. Consider the 1D-problem $U = [-1, 1]$, $||\nabla u|| - 1 = |u_x| - 1 = F(\nabla u)$ with the boundary conditions $u(-1) = u(1) = 0$. This problem allows infinitely many piecewise linear, weakly differentiable solutions, however no solution in the classical sense can be found for this problem. ♦

We can see that for the problem in Example 2.4, we can find no solutions in the classical sense to (2.23). It is still important to study problems of this kind, which is why we need to introduce the more permissive notion of generalized/weak solutions:

Theorem 2.2 (Generalized/Weak solutions)**.** *A weak solution u of* (2.21) *is Lipschitz continuous ($u \in C^{0,1}(\overline{U})$) on the closure $\overline{U}$, assumes the prescribed boundary values at ∂U and satisfies* (2.21) *almost everywhere.*

Note that the strong similarity between Theorem 2.2 and Definition 2.7 shows that the presented Theorem could also be expressed using weak derivatives. With this new definition, we can now find weak solutions of our previous example:

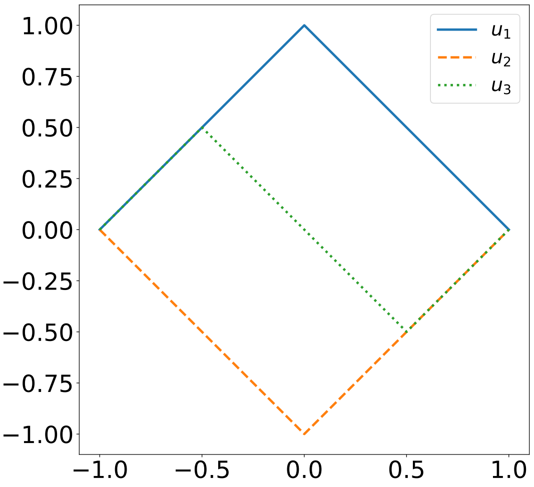


Figure 2.6: Three different functions u_1 through u_3 that are weak solutions of Example 2.4.

Example 2.4. [cont.] Consider now the three functions $u_1(x) = |x| - 1$, $u_2(x) = 1 - |x|$ and

$$u_3(x) = \begin{cases} x - 1 & \text{if } x < -0.5 \\ -x & \text{if } |x| < 0.5 \\ x - 1 & \text{else} \end{cases}$$

visualized in Figure 2.6. Clearly all three of these function are weak solutions to the considered problem, meaning they satisfy $|u_x| - 1$ in all but singular points (almost everywhere).

We presented several weak solutions to the problem, but it is immediately obvious this notion of solution is too weak and non-unique for the problem at hand. The desired special class of solutions we are interested in are called *viscosity solutions*, a subclass of weak solutions [19, Chapter 3], [43]. We will use here the super-/subdifferentials $\nabla^{\pm}u(\mathbf{x})$ again (see Definition 2.5):

Definition 2.16. *A function $u \in C^{0,1}(U)$ is a first order only* viscosity subsolution *of (2.21) if*

$$F(p, u(\mathbf{x}), \mathbf{x}) \leq 0 \quad \forall \mathbf{x} \in U, \mathbf{p} \in \nabla^{+}u(\mathbf{x}) \tag{2.24}$$

and similarly, a viscosity supersolution *can be defined as*

$$F(p, u(\mathbf{x}), \mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in U, \mathbf{p} \in \nabla^{-}u(\mathbf{x}) \tag{2.25}$$

We then say that u is a viscosity solution if both (2.24) and (2.25) hold, i.e. u is both a viscosity sub- and supersolution.

With the notion of viscosity solutions, we can revisit Example 2.4 (again):

Example 2.4. [cont.] We test now u_1 through u_3 to see if they are in fact viscosity solutions. In fact, we only need to test the non-differentiable points, since $\{\nabla\} = \nabla^+ = \nabla^-$ for points where the continuous derivative exists. We can see that $\nabla^- u_3(0.5) = \nabla^- u_2(0) = [-1, 1]$, for which both $F(0 \in \nabla^- u_2(0), u, 0) = |0| - 1 \ngeq 0$ and $F(0 \in \nabla^- u_3(0.5), u, 0.5) = |0| - 1 \ngeq 0$ violate the viscosity supersolution condition. In contrast $\nabla^- u_1(0) = \emptyset$ and $\nabla^+ u_1(0) = [-1, 1]$, which satisfies $F(p \in \nabla^+ u_2(0), u, 0) \leq 0$. Hence u_1 is the sought viscosity solution to our problem.

Non-differentiability of the eikonal solution will also later appear in Chapter 4, similar to Example 2.4. We often refer to these areas of non-differentiability as front-collisions. In more mathematical terms, this simply refers to areas that are equidistant to the boundary according to the distance measure D, induced on the domain U. From here on out, every time we refer to an eikonal solution, we in fact refer to the viscosity solution of the problem.

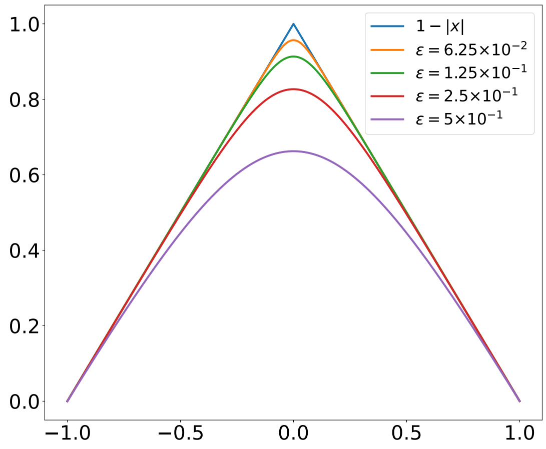


Figure 2.7: Method of vanishing viscosity, from which the name viscosity solutions originates, for the problem presented in Example 2.4. Instead of directly solving $|u_x| - 1 = 0$, we solve $u_x^2 - 1 = \varepsilon u_{xx}$. As $\varepsilon \to 0$, we converge to the desired viscosity solution.

Remark 2.3. *An alternative approach to introduce these solutions is by considering the original problem, coupled with a vanishing viscosity term $F(\nabla u(\mathbf{x}), u(\mathbf{x}), \mathbf{x}) = -\varepsilon \Delta u(\mathbf{x})$ [49]. As the viscosity vanishes ($\varepsilon \to 0$), we also arrive at the solution u_1 to the problem, hence the name "viscosity solution". The solutions of Example 2.4 using this method as $\varepsilon \to 0$ can be seen in Figure 2.7.*

We now consider a popular initial-value problem called the Hamilton–Jacobi equation:

$$\begin{aligned} u_t + H(\nabla u) &= 0 \quad \text{in } \mathbb{R}^n \times (0, \infty) \\ u &= g \quad \text{on } \mathbb{R}^n \times \{t = 0\} \end{aligned} \tag{2.26}$$

With $u : \mathbb{R}^n \times [0, \infty)$ being the unknown again $u = u(\mathbf{x}, t)$ and $\nabla u = \nabla_{\mathbf{x}} u$. $H : \mathbb{R}^n \to \mathbb{R}$ being the *Hamiltonian* and for a given initial function $g : \mathbb{R}^n \to \mathbb{R}$. Using a change of variable $\nabla u = \mathbf{p}$, the associated characteristic equation to (2.26) then becomes

$$\begin{cases} \dot{\mathbf{x}} = \nabla_{\mathbf{p}} H(\mathbf{p}, \mathbf{x}) \\ \dot{\mathbf{p}} = -\nabla_{\mathbf{x}} H(\mathbf{p}, \mathbf{x}). \end{cases} \tag{2.27}$$

Additionally, we introduce the corresponding *Lagrangian* function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ $\mathcal{L} = \mathcal{L}(\dot{\gamma}(t), \gamma(t)) = \mathcal{L}(\mathbf{v}, \mathbf{x})$, which in the case of the eikonal equation becomes

$$\mathcal{L}(\mathbf{v}, \mathbf{x}) = ||\mathbf{v}||_{D^{-1}(\mathbf{x})} . \tag{2.28}$$

Using this definition, we can define the length of an arbitrary curve over time $[0, T]$ for $T > 0$ as:

$$L(\gamma) := \int_0^T \mathcal{L}(\dot{\gamma}(t), \gamma(t)) \, \mathrm{d}t. \tag{2.29}$$

Most commonly, we are interested in fixed curves starting from $\mathbf{x} \in \mathbb{R}^n$ at time $t = 0$ and ending in $\mathbf{y} \in \mathbb{R}^n$ at $t = T$. The set of admissible curves fulfilling this criterion can described by the set:

$$\mathcal{A} := \left\{ \gamma \in C^{0,1}([0, t]; \mathbb{R}^n) | \gamma(0) = \mathbf{x}, \gamma(t) = \mathbf{y} \right\} .$$

This admits the notion of length-minimizing paths $\widehat{\gamma}$, called geodesics:

$$\widehat{\gamma} = \arg\min_{\gamma \in \mathcal{A}} L(\gamma). \tag{2.30}$$

The associated length of the geodesic path $\delta(\mathbf{x}, \mathbf{y}) := L(\widehat{\gamma})$ is called *geodesic distance*. These geodesics in our induced metric D have several interesting properties:

- They solve the Euler–Lagrange system of equations for all times $t \in [0, T]$, given by

$$-\frac{\partial}{\partial t} \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}(t), \widehat{\gamma}(t)) + \nabla_{\widehat{\gamma}} \mathcal{L}(\mathbf{v}(t), \widehat{\gamma}(t)) = 0 \tag{2.31}$$

- Under the assumption that $D \in C^1(\mathbb{R}^n, S^n_{++})$, the distance δ also ensures the properties of a norm introduced in Section 2.2, most importantly the triangle inequality $(\delta(\mathbf{x}, \mathbf{z}) \leq \delta(\mathbf{x}, \mathbf{y}) + \delta(\mathbf{y}, \mathbf{z}))$.
- Curves γ that solve the Euler–Lagrange system of equations, but are no minimizers of (2.30) are sometimes called critical points.

2.8.2 Discretization

Up until now, we considered that a classical or viscosity solution for a given PDE in (2.21) can be found analytically. Many practical applications however require that we find an approximative solution to the problem for any given domain, especially in use-cases where the domains originate from real world measurements and objects (e.g. 3D scans, structural analysis, machine designs). In this scenario, it is infeasible to construct analytical solutions for each of these cases.

For this purpose, we revisit a commonly encountered definition of the directional derivative of $f : \mathbb{R}^n \to \mathbb{R}$ in direction $\mathbf{v} \in \mathbb{R}^n$:

$$\langle \nabla f(\mathbf{x}), \mathbf{v} \rangle = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h} \tag{2.32}$$

From (2.32), we can see that a straight-forward approximation of $\langle \nabla f(\mathbf{x}), \mathbf{v} \rangle f$ can be achieved by computing

$$\langle \nabla f(\mathbf{x}), \mathbf{v} \rangle = \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h} + \mathcal{O}(h) \approx \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h}, \tag{2.33}$$

for a small enough h. Note that the full gradient $\nabla f(\mathbf{x})$ can be retrieved using (2.32) as:

$$\nabla f(\mathbf{x}) = \left(\langle \nabla f(\mathbf{x}), \mathbf{e}_1 \rangle, \ldots, \langle \nabla f(\mathbf{x}), \mathbf{e}_n \rangle \right)^\top$$

for $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{e}_i \in \mathbb{R}^n$ being the canonical basis vectors.

Consider now a rectangular domain $[a, b]^n$ with evenly distributed points $\mathbf{x}_{i_1,\ldots,i_n}$ inside (i.e. an uniform grid), each of which has a distance of h to its next neighbors as shown in Figure 2.8 for $n = 2$. We can see that we can apply (2.33) naturally in this formulation to approximate ∇f, by taking the pointwise differences

$$\nabla f(\mathbf{x}_{i_1,\ldots,i_n}) = \frac{1}{h} \left(\begin{pmatrix} f(\mathbf{x}_{i_1+1,\ldots,i_n}) \\ \vdots \\ f(\mathbf{x}_{i_1,\ldots,i_n+1}) \end{pmatrix} - \begin{pmatrix} f(\mathbf{x}_{i_1,\ldots,i_n}) \\ \vdots \\ f(\mathbf{x}_{i_1,\ldots,i_n}) \end{pmatrix} \right), \tag{2.34}$$

where we need to enforce boundary conditions on the boundary.

This method is called the Finite Difference Methods (FDMs) [90] and allows to produce efficient computational models, especially for rectangular domains. Note that the chosen discretization in (2.33) and (2.34) is just one of many possibilities on how to discretize the FDs. The chosen discretization is only first order accurate ($\mathcal{O}(h^1)$ in (2.33)), but many more schemes have been proposed with higher order accuracy [50], which require to look at a bigger neighborhood around the current point.

While computationally efficient, the FDM is still not an optimal tool for arbitrary domains and needs additional conditions to be used with curved boundaries for instance. Another very popular method is the FEM, discussed in the following section.

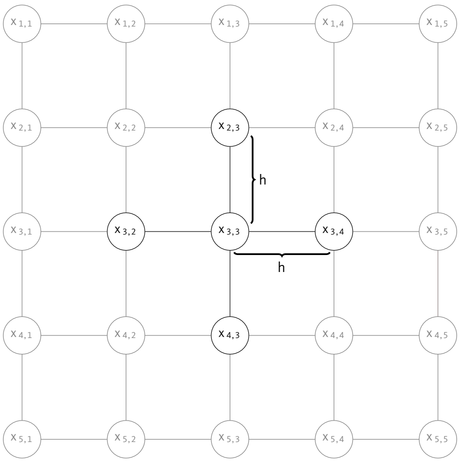


Figure 2.8: Visualization of a typically employed uniform grid to efficiently compute FDs in 2D. The sampled points $\mathbf{x}_{i,j}$ are equidistant to each other and the gradient of a quantity defined at each point can be approximated taking the difference along the row and column. Note that more general formulations allow for different spacings for each dimension or in certain regions of the domain.

2.9 Finite Element Method (FEM)

Undoubtedly, FEMs are one of the major cornerstones of modern modeling and simulation in many engineering disciplines. We briefly introduced the classical solution of PDE problems in Section 2.8, discussed that finding classical solutions for general domains and discretizations is a non-trivial task and showed the FDM as one solution in Section 2.8.2. To obtain solutions for more complicated domains — ill-suited for the FDM —, we decompose the domain into a smaller subset of elements (hence the name finite elements) on each of which an analytical solution can be obtained with a certain degree of continuity assured on the global solution (piecewise polynomial). We already introduced our three used concepts of *classical*, *weak* and *viscosity* solutions in Section 2.8. However, we did not specifically mention how to efficiently compute weak solutions to PDE problems. This is the point where FEMs come in handy. This section is heavily based on the books [18, 49, 82].

Definition 2.17. *Consider the following three building blocks of finite elements:*

1. *The* element domain $\Omega_E \subset \mathbb{R}^n$ *is given as a bounded closed set with nonempty interior and piecewise smooth boundary*
2. *The finite-dimensional space of* basis functions *(sometimes also called shape functions)* $\mathcal{P}$ *on* Ω_E
3. *The set of* nodal variables $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$ *is a basis for the dual space* $\mathcal{P}'$

The triplet $(\Omega_E, \mathcal{P}, \mathcal{N})$ *is then called a Finite Element (FE).*

More precisely, the basis $\{\phi_1, \phi_2, \ldots, \phi_k\}$ of $\mathcal{P}$ is dual to $\mathcal{N}$, for which the Kronecker property $N_i(\phi_j) = \delta_{i,j}$ holds. Condition 3 of Definition 2.17, can be rewritten in a form that is easier to verify [18]:

Definition 2.18. *Let $\mathcal{P}$ be a d-dimensional vector space let $\{N_1, \ldots, N_d\}$ be a subset of the dual space $\mathcal{P}'$. Then condition 3 of Definition 2.17 is equivalent to: Given $v \in \mathcal{P}$ with $N_i v = 0$ for $i = 1, \ldots, d$, then $v = 0$.*

Example 2.5. [Lagrange $\mathcal{P}_1$ element in $\mathbb{R}^2$] Consider the domain $\Omega_E = \{(x,y)^\top | x > 0 \wedge y > 0 \wedge x + y \leq 1\}$, with $\mathcal{P}$ being the set of linear polynomials and $\mathcal{N} = \{N_1, N_2, N_3\}$ as the nodal basis, where $N_1(v) = v(0,0)$, $N_2(v) = v(1,0)$ and $N_3(v) = v(0,1)$. We call $(\Omega_E, \mathcal{P}, \mathcal{N})$ in this case the Lagrange $\mathcal{P}_1$ element in 2D with the associated nodal basis $\phi_1(x,y) = 1 - x - y$, $\phi_2(x,y) = x$ and $\phi_3(x,y) = y$.

While the first two conditions of Definition 2.17 are easily verified, we can check the validity of the last condition by using the Definition 2.18. $v \in \mathcal{P}_1$ implies $v = a + bx + cy$, whereas $N_i(v) = 0$ implies $v = 0$ and thus $\mathcal{N}$ is a valid basis for $\mathcal{P}$. ⧫

We will heavily make use the element presented in Example 2.5. Note that the origin of the nodal bases functions are sometimes called Degrees of Freedoms (DOFs). However, the definition in Example 2.5 only describes the uniform, axis-oriented triangle located at the origin. Consider a domain composed of several FEs forming the domain $\Omega = \bigcup_i \Omega_{E_i}$. Defining a FE for e.g. every possible triangle of a triangular mesh can quickly become an intractable task, or would require the definition of a general element with spatially varying DOFs. We instead use a method to generalize chosen reference elements to general elements of the same type on the domain: Consider the reference element $\tilde{\Omega}_E$ with the nodal bases $\mathcal{N} = \{\tilde{N}_1, \ldots \tilde{N}_d\}$ and given DOFs $\tilde{\mathbf{v}}_i \in \tilde{\Omega}_E$. Each point in the reference element $\tilde{\mathbf{x}} \in \tilde{\Omega}_E$ can be defined as $\tilde{\mathbf{x}} = \sum_i \tilde{N}_i(\tilde{\mathbf{x}})\tilde{\mathbf{v}}_i$. Now consider a general element with a point inside $\mathbf{x} \in \Omega_E$. We can express said point as a function of its corresponding point on the reference element $\tilde{\mathbf{x}}$, i.e. $\mathbf{x} : \tilde{\Omega}_E \to \Omega_E$. As both elements use the same polynomial basis $\mathcal{P}$, the corresponding function is given by

$$\mathbf{x}(\tilde{\mathbf{x}}) = \sum_{i=1}^{d} \tilde{N}_i(\tilde{\mathbf{x}})\mathbf{v}_i,$$

for $\mathbf{v}_i \in \Omega_E$ being the DOFs of the real element. For details on why this is possible, we refer to [18]. This mapping allows us to define the Jacobi matrix $J_{\mathbf{x},\tilde{\mathbf{x}}}$ of the reference to the real coordinate system as [82]:

$$J_{\mathbf{x},\tilde{\mathbf{x}}} = \begin{pmatrix} \frac{\partial x_1}{\partial \tilde{x}_1} & \cdots & \frac{\partial x_n}{\partial \tilde{x}_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial \tilde{x}_n} & \cdots & \frac{\partial x_n}{\partial \tilde{x}_n} \end{pmatrix} = \sum_{i=1}^{d} \mathbf{v}_i \otimes \nabla_{\tilde{\mathbf{x}}} \tilde{N}_i \tag{2.35}$$

This allows us to easily map functions, or derivative of functions from the reference element to the real element, or vice versa:

$$\left(J_{\mathbf{x},\tilde{\mathbf{x}}}^{-1}\right)^{\top} \nabla_{\tilde{\mathbf{x}}} f = \begin{pmatrix} \frac{\partial \tilde{x}_1}{\partial x_1} & \cdots & \frac{\partial \tilde{x}_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \tilde{x}_1}{\partial x_n} & \cdots & \frac{\partial \tilde{x}_n}{\partial x_n} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial \tilde{x}_1} \\ \vdots \\ \frac{\partial f}{\partial \tilde{x}_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} = \nabla_{\mathbf{x}} f \tag{2.36}$$

This mapping for a $\mathcal{P}_1$ element in 2D is visualized in Figure 2.9. Note that the inverse in (2.36) exists for non-degenerate elements, which are the only type considered in this thesis.

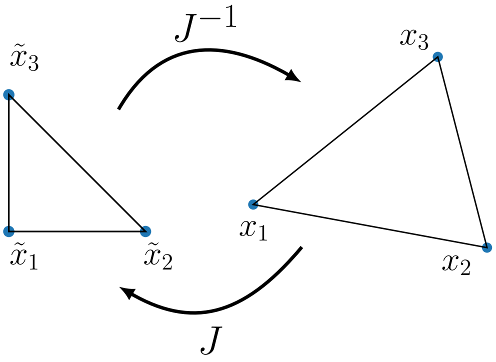


Figure 2.9: Using $J_{\mathbf{x},\tilde{\mathbf{x}}}$, we can map points from a defined reference element to a real element in the mesh. In the special case of Lagrange $\mathcal{P}_1$ elements, J is a piecewise constant function across the domain/mesh.

Besides the presented Lagrangian P1 FE space, a plethora of different FEs have been presented and promoted. We conclude the discussion about different element types with Figure 2.10 showing the infamous periodic table of finite elements, which can be found at `http://www-users.math.umn.edu/~arnold/femtable/`, showing the wide range of possible element discretizations. Note that this work will solely rely on Lagrangian P1 FEs (triangles or tetrahedra) for all instances where we need to use FEM.

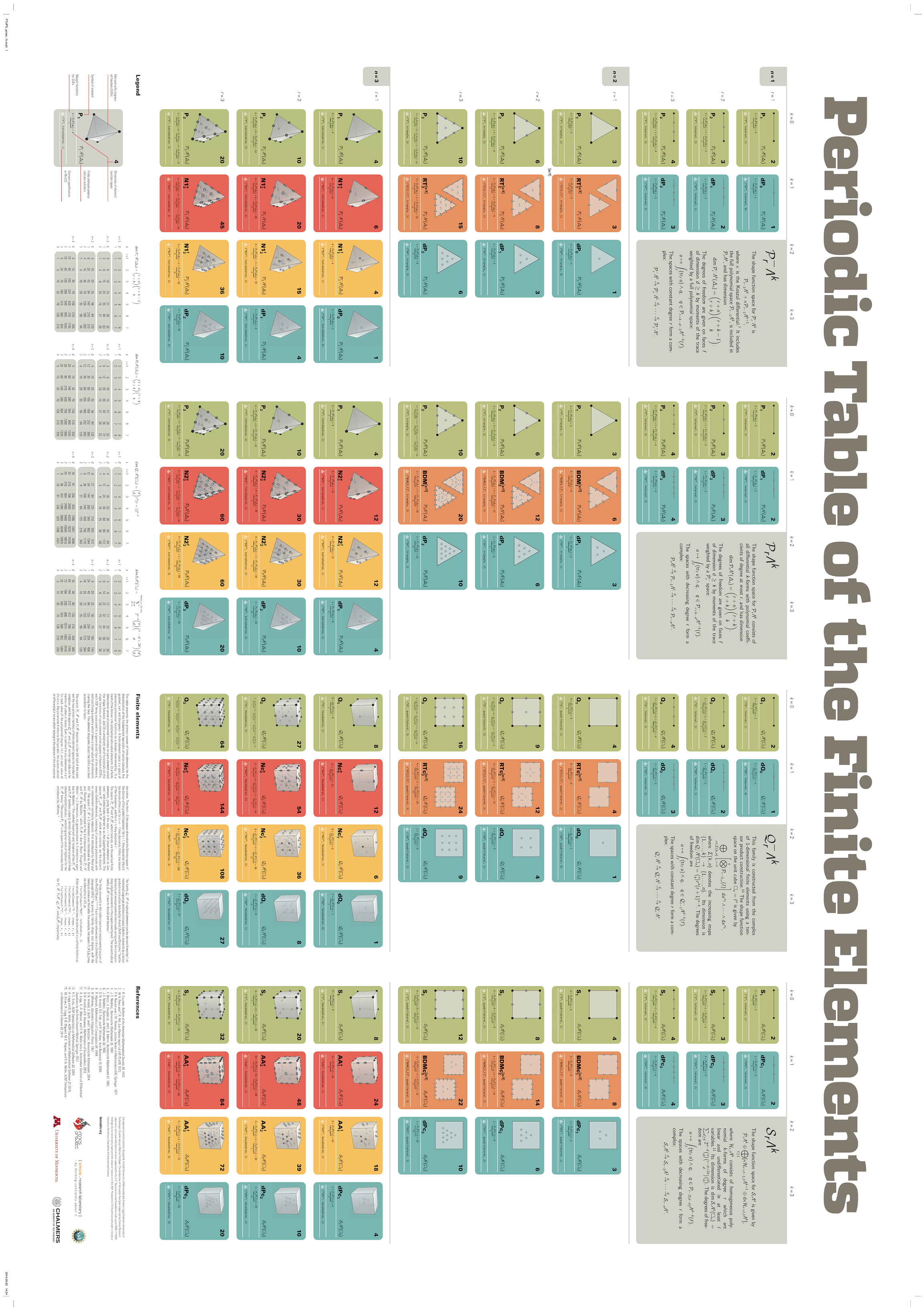


Figure 2.10: The infamous periodic table of finite elements, found on `http://www-users.math.umn.edu/~arnold/femtable/`

Using these FEs, we can build a linear solution both to perform L^2-projections of known functions on the mesh, as well as solve PDEs in a weak sense. For this purpose, we seek to compute the nodal values u_i, associated to each nodal basis N_i $(i = 1, \dots, n)$ such that it solves an equation in a weak, or variational sense. Consider for this purpose a function $f : L^2(\Omega) \to \mathbb{R}$ to be approximated, such that

$$f(\mathbf{x}) \approx u_h(\mathbf{x}) = \sum_j N_j(\mathbf{x}) u_j$$

where u_h is a sought continuous, piecewise polynomial function. Our objective is then to find

$$\min_{u_h \in V_h} \frac{1}{2} \int_\Omega (u_h(\mathbf{x}) - f(\mathbf{x}))^2 \, \mathrm{d}\mathbf{x}, \tag{2.37}$$

where V_h is the space of continuous, piecewise polynomial functions. The weak/variational formulation of this problem then reads

$$\begin{aligned} \int_\Omega f(\mathbf{x}) v(\mathbf{x}) \, \mathrm{d}\mathbf{x} &= \int_\Omega u_h(\mathbf{x}) v(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ \int_\Omega f(\mathbf{x}) N_i(\mathbf{x}) \, \mathrm{d}\mathbf{x} &= \int_\Omega \sum_j N_j(\mathbf{x}) u_j N_i(\mathbf{x}) \, \mathrm{d}\mathbf{x} \quad \text{for } i = 1, \dots, n \end{aligned}$$

where the transition from v to the basis functions N_i follows from the fact that if the problem is satisfied for each N_i, then it is also satisfied for their linear combination reflected by v. By introducing the (sparse) mass matrix $M_{i,j} = \int_\Omega N_i(\mathbf{x}) N_j(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ and $b_i = \int_\Omega N_i(\mathbf{x}) f(\mathbf{x})$, we can rewrite the problem as:

$$Mu = b. \tag{2.38}$$

The integrals $\int_\Omega f(\mathbf{x})$ can be approximated using a Gauss-quadrature rule, which reads as $\sum_i w_i f(\mathbf{x}_i)$ for specified points $\mathbf{x}_i \in \Omega$ and associated weights $w_i \in \mathbb{R}$. In practice, all integrals are actually evaluated only on the reference element and the areas and values are mapped to each point using the Jacobi matrix in (2.36). The determinant of the Jacobi matrix gives information on the change of the integral between the two domains.

Similarly, we can formulate linear PDEs also as a linear system to be solved, just as we did for the L_2-projection in (2.38): We will demonstrate this on Poisson's equation in (2.22):

Example 2.6. [Poisson's equation] Recall Poisson's equation:

$$\begin{aligned} -\Delta u = f, \ \text{in } \Omega \\ \langle \nabla u, \mathbf{n} \rangle = 0, \ \text{on } \partial\Omega \end{aligned}$$

We reformulate the problem in variational sense using the previously discussed concepts and by applying Green's formula/identity:

$$\begin{aligned}\int_\Omega f(\mathbf{x})v(\mathbf{x})\,\mathrm{d}\mathbf{x} &= -\int_\Omega \Delta u(\mathbf{x})v(\mathbf{x})\,\mathrm{d}\mathbf{x}\\ &= \int_\Omega \langle \nabla u(\mathbf{x}), \nabla v(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x} - \int_{\partial\Omega} \langle \mathbf{n}(\mathbf{x}), \nabla u(\mathbf{x})\rangle\, v(\mathbf{x})\,\mathrm{d}\mathbf{x}\\ &= \int_\Omega \langle \nabla u(\mathbf{x}), \nabla v(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x}\end{aligned}$$

which similar to (2.9) can be defined by the use of our nodal basis functions:

$$\begin{aligned}\int_\Omega f(\mathbf{x})N_i(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_\Omega \langle \nabla u_h(\mathbf{x}), \nabla N_i(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x} \quad \text{for } i = 1,\dots,n\\ &= \int_\Omega \left\langle \nabla \sum_j N_j(\mathbf{x})u_j, \nabla N_i(\mathbf{x})\right\rangle\mathrm{d}\mathbf{x}\\ &= \sum_j u_j \int_\Omega \langle \nabla N_j(\mathbf{x}), \nabla N_i(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x}\end{aligned}$$

By defining the stiffness matrix $A_{i,j} = \int_\Omega \langle \nabla N_i(\mathbf{x}), \nabla N_j(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x}$ and $b_i = \int_\Omega f(\mathbf{x})N_i(\mathbf{x})$, we can again write the problem simply as

$$A\mathbf{u} = b \tag{2.39}$$

♦

Remark 2.4. *Since the nodal basis function N_i have compact support, the resulting mass and stiffness matrices M and A will generally be very sparse. Rather than evaluating the integrals over the whole domain, it is therefore beneficial to integrate element-wise, where only elements are to be considered that are in the support of the nodal basis function.*

Note that the use of the term projection for (2.37) slightly differs from the later used definition in Section 2.10, but is commonly encountered when talking about FEM.

2.10 Optimization

Up until this point, we only considered well-posed linear problems of the type:

$$A\mathbf{x} = \mathbf{b} \tag{2.40}$$

where $A \in \mathbb{R}^{n\times n}$ is assumed to be a square and invertible matrix (i.e. $\operatorname{rank}(A) = n$) and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ are the unknown and given vectors respectively. Such problems can be viewed as the optimal solutions to minimization problems of the type

$$\min_{\mathbf{x}} \frac{1}{2}\langle \mathbf{x}, A\mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle\,, \tag{2.41}$$

and can either be explicitly solved by computing the inverse A^{-1} e.g. through means of the QR-decomposition, or with an iterative solver such as conjugate gradient method [69].

However, consider the case where we are interested in solving (2.40), even though A is either a non-square matrix (i.e. the system is over- oder undertermined) or does not have full rank. We can achieve this by replacing the original problem with a minimization problem

$$\min_{\mathbf{x}} \frac{1}{2} \left|\left|A\mathbf{x} - \mathbf{b}\right|\right|_2^2 . \tag{2.42}$$

In order to better understand how we could efficiently solve (2.42) and what properties this equation has, it is therefore worth studying optimization problems in general.

2.10.1 Convex Optimization Problems

We now consider more general constrained optimization problems of the form

$$\min_{\mathbf{u}} f(\mathbf{u}) \text{ s.t. } \mathbf{u} \in C \tag{2.43}$$

for a convex set $C \subset \mathbb{R}^n$. In order to define convex sets, we define them through the definition of affine sets [13, p. 3]

Definition 2.19 (Affine and Convex Sets)**.** *Given a subset $S \subseteq \mathbb{R}^n$ is called affine if for any two points $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in \mathbb{R}$, their affine combination is contained in S, i.e.:*

$$\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in S. \tag{2.44}$$

A set $C \subseteq \mathbb{R}^n$ is called convex, if for any two points $\mathbf{x}, \mathbf{y} \in C$ and $\lambda \in [0, 1]$, it holds that

$$\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in C. \tag{2.45}$$

From Definition 2.19, we can see that all affine sets are also convex. Note that $\mathbb{R}^n$ is a convex set. Commonly encountered convex sets in this thesis include unit norm-balls $||\cdot|| \leq 1$, the set of symmetric positive definite matrices S^n_{++} in (2.6) (depending on their parameterization) and the simplex in n dimensions

$$\left\{ \sum_{i=1}^{n+1} \lambda_i \mathbf{v}_i \,\middle|\, \sum_{i=1}^{n+1} \lambda_i = 1 \wedge \lambda_i \geq 0, \, i = 1, \ldots, n \right\}, \tag{2.46}$$

defined by its vertices/corners $\mathbf{v}_i \in \mathbb{R}^n$.

In its general form, it is difficult to associate any properties of how difficult (2.43) is to solve, but we can further analyze the problem if we introduce the notion of convex functions. The considerations on how to optimize such general functions are thoroughly discussed in [13, 102], on which this section is based on. We use several different, yet equal properties that define a convex function f:

Definition 2.20 (Convex Functions)**.** *For a function $f : \mathbb{R}^n \supset U \to \mathbb{R}$, the following statements regarding convexity are equivalent*

(a) f is a convex function
(b) The epigraph of f, given by epi$(f) = \{(\mathbf{x}, y) | f(\mathbf{x}) \leq y, \mathbf{x} \in U, y \in \mathbb{R}\}$, is a convex set
(c) $\forall \lambda \in [0, 1], \forall \mathbf{x}, \mathbf{y} \in U : f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$
(d) If additionally $f \in C^1$, then $\forall \mathbf{x}, \mathbf{y} \in U : f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$
(e) If additionally $f \in C^2$, then the Hessian $\nabla^2 f(\mathbf{x})$ is positive semi-definite on U, i.e. $\forall \mathbf{x} \in U : \nabla^2 f \in \bar{S}^n_{++}$ (see (2.7))

Note that (c) is a special case of Jensen's inequality for convex functions

$$f\left(\sum_{i=1}^{k} \lambda_i \mathbf{x}_i\right) \leq \sum_{i=1}^{k} \lambda_i f(\mathbf{x}_i), \tag{2.47}$$

for $\mathbf{x}_1, \ldots, \mathbf{x}_k \in U$ and $\sum_i \lambda_i = 1 \wedge \lambda_i \geq 0$.

When solving (2.43), we want to find the global minimizer $\mathbf{x}^*$, such that

$$\forall \mathbf{x} \in C : f(\mathbf{x}^*) \leq f(\mathbf{x}).$$

However, in practice (at least for non-convex problems) we are often content with finding a local minimizer $\mathbf{x}^*$, for which

$$\forall \mathbf{x} \in N_\varepsilon : f(\mathbf{x}^*) \leq f(\mathbf{x}),$$

where N_ε is a local neighborhood around a point $\mathbf{x}$. In gradient based optimization, this condition is equal to the optimality condition (sometimes called Fermat's theorem of stationary point) that we often employ for $f \in C^1$

$$0 = \nabla f(\mathbf{x}^*). \tag{2.48}$$

For weakly differentiable functions, we have to extend our previous definition of subdifferentials in (2.11) to support the function f on the whole domain U [102, p. 300 ff]:

$$\partial f(\mathbf{x}) = \{\mathbf{v} \in \mathbb{R}^n \,|\, f(\mathbf{z}) \geq f(\mathbf{x}) + \langle \mathbf{v}, \mathbf{z} - \mathbf{x} \rangle \quad \forall \mathbf{z} \in U\} \tag{2.49}$$

Note that this global subdifferential is distinctively denoted by ∂ instead of ∇^-.

The optimality condition for weakly differentiable functions $f \in H^1$ (see Definition 2.9) then becomes:

$$0 \in \partial f(\mathbf{x}^*). \tag{2.50}$$

Note that for convex functions, (2.48) and (2.50) ensure that $\mathbf{x}^*$ is a global minimizer.

We previously already defined Lipschitz continuous functions in Definition 2.6, but in the context of convex functions it is beneficial to also look at functions with Lipschitz continuous gradients [13, p. 110 ff]:

Definition 2.21 (Lipschitz continuous gradients)**.** *Let $L \geq 0$. The function $f \in C^1(\mathbb{R}^n, \mathbb{R})$ is said to have Lipschitz continuous gradient with parameter L, if it satisfies*

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})|| \leq L\,||\mathbf{x} - \mathbf{y}|| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{2.51}$$

Definition 2.21 resembles Definition 2.6 on the gradients of the function and is particularly of importance for convex optimization. Note that if a function has a Lipschitz continuous gradient of L_1, then the function also has a Lipschitz continuous gradient of L_2 if $L_2 \geq L_1$. We are mostly interested in the smallest possible L, either over the whole domain, or for a restricted neighborhood N_ε around the current point. We will call the former L global Lipschitz constant of the gradient, while the latter will be called the local Lipschitz constant of the gradient. Lipschitz continuous gradients are particularly useful for convex functions:

Theorem 2.3. *For a convex function f, the following statements are equivalent:*

1. *f has a Lipschitz continuous gradient L*
2. *If additionally $f \in C^1$, then $f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{L}{2}\,||\mathbf{x} - \mathbf{y}||^2$ (Descent Lemma)*
3. *$f(\lambda\mathbf{x} + (1 - \lambda\mathbf{y})) \geq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y}) - \frac{L}{2}\lambda(1 - \lambda)\,||\mathbf{x} - \mathbf{y}||^2$ for $\lambda \in [0, 1]$.*

Especially the descent lemma is very useful as this is used later in Chapter 4 for both the Lipschitz backtracking and the (F)ISTA algorithm [12].

A common method to optimize (2.43) for a function $f \in C^1$ with no constraints is the well known gradient/steepest descent algorithm, which starts from an initial parameter guess $\mathbf{x}_0 \in \mathbb{R}^n$ and updates the parameters in each iterations as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla_{\mathbf{x}} f(\mathbf{x}_k), \tag{2.52}$$

for $t_k > 0$. This solution strategy is a consequence of using the previously mentioned Euler-forward scheme in (2.15). The descent lemma confirms that the negative gradient direction is a descent direction $\mathbf{p}_\varepsilon = -\varepsilon \nabla_{\mathbf{x}} f$ for which

$$\langle \mathbf{p}, \nabla_{\mathbf{x}} f \rangle < 0.$$

It is therefore used to decrease the function, i.e. $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ for a small enough step size t_k (see [13, p. 196 ff]). In Chapter 4 we will often set $t_k = L^{-1}$ – a consequence of applying the descent lemma around a point $\mathbf{x}_k$:

$$\mathbf{x} = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k) = \arg\min_{\mathbf{x}} f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{L}{2}\left|\left|\mathbf{x} - \mathbf{x}_k\right|\right|_2^2 \tag{2.53}$$

The similar subgradient descent algorithm can also be used to optimize function that are merely Lipschitz continuous, i.e. $f \in C^{0,1}$. In this case, we replace the gradient ∇f in (2.52) with an element of the subdifferential $\mathbf{p} \in \partial f$ from (2.49). Note however that we lose the guarantee that the subgradient is a descent direction [13, p. 196].

2.10.2 Non-smooth Optimization

An alternative approach to optimize such non-smooth functions is to use the proximal operator, or mapping:

Definition 2.22 (Proximal Mapping). *Consider a function $f \in \mathbb{R}^n \to \mathbb{R}$, then the proximal mapping of the function f is given by:*

$$\mathbf{x} = \left(\mathbf{I} + \tau\partial f\right)^{-1}(\tilde{\mathbf{x}}) = \underset{\tau f}{\mathrm{prox}}(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}} \tau f(\mathbf{x}) + \frac{1}{2}\left|\left|\mathbf{x} - \tilde{\mathbf{x}}\right|\right|_2^2 \tag{2.54}$$

Note that $\mathbf{x}$ in Definition 2.22 exists and is additionally uniquely defined if f is a proper closed and convex function. A property of proximal mappings that we will use regularly throughout this work is the linear separability.

Theorem 2.4. *Consider a function $f : \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_d} \to \mathbb{R}$, given by*

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_d) = \sum_{i=1}^{d} f_i(\mathbf{x}_i),$$

then the proximal mapping can be expressed as

$$\underset{\tau f}{\mathrm{prox}}(\mathbf{x}_1, \ldots, \mathbf{x}_d) = \underset{\tau f_1}{\mathrm{prox}}(\mathbf{x}_1) \times \ldots \times \underset{\tau f_d}{\mathrm{prox}}(\mathbf{x}_d)$$

i.e. the proximal mappings can be applied independently to each component $\mathbf{x}_i$.

Example 2.7. [Proximal Mapping of the l_1 norm ($\mathrm{prox}_{\tau||\cdot||_1}$)] Consider the non-smooth l_1 norm, $f(\cdot) = ||\cdot||_1 = \sum|\cdot|$. We can then calculate the proximal mapping $\mathrm{prox}_{\tau f}$ and its gradient as

$$\underset{\tau f}{\mathrm{prox}}(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}} \tau f(\mathbf{x}) + \frac{1}{2}\left|\left|\mathbf{x} - \tilde{\mathbf{x}}\right|\right|_2^2$$
$$0 \in \tau\partial f(\mathbf{x}) + \mathbf{x} - \tilde{\mathbf{x}}.$$

Since the problem is linearly separable, we can solve the problem for each vector component independently. Specifically, we have to solve for the three cases:

$$\begin{aligned}
x_i > 0: \quad & 0 && = \tau + x_i - \tilde{x}_i \\
& x_i && = \tilde{x}_i - \tau \Rightarrow \tilde{x}_i > \tau \\
x_i < 0: \quad & 0 && = -\tau + x_i - \tilde{x}_i \\
& x_i && = \tilde{x}_i + \tau \Rightarrow \tilde{x}_i > -\tau \\
x_i = 0: \quad & 0 && \in \tau\,[-1,1] - \tilde{x}_i \\
& \tilde{x}_i && \in [-\tau, \tau]
\end{aligned}$$

The resulting function

$$\left(\underset{\tau f}{\mathrm{prox}}(\mathbf{x})\right)_i = \mathrm{sgn}(x_i)\max(0, |x_i| - \tau) = \begin{cases} x_i - \tau & \text{if } x_i > \tau \\ x_i + \tau & \text{if } x_i < -\tau \\ 0 & \text{else} \end{cases}, \tag{2.55}$$

visualized in Figure 2.11 is usually referred to as the softshrinkage function and commonly used in many optimization algorithms. ◆

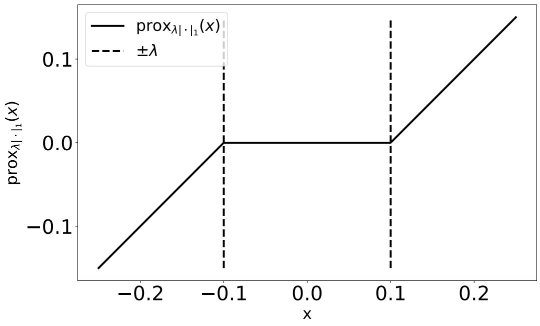


Figure 2.11: The softhsrinkage function $\mathrm{prox}_{\tau|\cdot|_1}(\mathbf{x})$

An useful class of functions are the convex conjugates f^* to a given function f [13, Chapter 4]:

Definition 2.23 (Convex Conjugates). *Let $f : U \to (-\infty, \infty]$. We call the function $f^* : U \to [-\infty, \infty]$ given by*

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in U} \{\langle \mathbf{y}, \mathbf{x} \rangle - f(\mathbf{x})\} \tag{2.56}$$

the convex conjugate of f.

A commonly encountered convex conjugate is that of norms, which is the indicator function on the unit-ball of their dual norm:

Lemma 1. *Consider an arbitrary norm $f(\mathbf{x}) = ||\mathbf{x}||$ for $\mathbf{x} \in U$. The convex conjugate f^* is then given by*

$$f^*(\mathbf{y}) = \delta_{||\cdot||_* \leq 1}(\mathbf{y}) = \begin{cases} 0 & \text{if } ||\mathbf{y}||_* \leq 1 \\ \infty & \text{else} \end{cases} \tag{2.57}$$

where $||\cdot||_*$ is the dual norm of $||\cdot||$.

With Definition 2.23, we can define the later very useful Moreau decomposition [13, p. 160 ff]:

Theorem 2.5 (Moreau Decomposition)**.** *Consider a closed convex function $f : U \to (-\infty, \infty]$ and its convex conjugate f^*. Then it holds for $\mathbf{x} \in U$ that*

$$\operatorname*{prox}_{\tau f}(\mathbf{x}) + \tau \operatorname*{prox}_{\frac{1}{\tau} f^*}(\mathbf{x}/\tau) = \mathbf{x}. \tag{2.58}$$

A special subclass of the previously introduced proximal mappings are projections. Consider now the indicator function on a convex set C

$$\delta_C(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in C \\ \infty & \text{else} \end{cases}.$$

The proximal mapping of such a function can be described as any of the following formulations [13, p. 146 ff]

$$\operatorname*{prox}_{\tau \delta_C}(\tilde{\mathbf{x}}) = \operatorname*{prox}_{\delta_C}(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}} \delta_C(\mathbf{x}) + \frac{1}{2} ||\mathbf{x} - \tilde{\mathbf{x}}||_2^2 = \arg\min_{\mathbf{x} \in C} \frac{1}{2} ||\mathbf{x} - \tilde{\mathbf{x}}||_2^2 = \operatorname*{proj}_C(\tilde{\mathbf{x}}), \tag{2.59}$$

where we usually use the last equality. For a closed and convex, non-empty set $C \subset U$, the projection $\text{proj}_C(\mathbf{x})$ for $\mathbf{x} \in U$ exists and is unique [12, p. 147]. Note that (2.59) is sometimes also called the orthogonal projection, since the closest projection in the Euclidean sense to a convex set C is the projection orthogonal to the surface of C ($\perp_{\mathbf{x}} C$). This can also be used to reconsider the previously introduced gradient descent algorithm in (2.52) and (2.53) in a constrained optimization setting:

$$\begin{aligned} \arg\min_{\mathbf{x}} f(\mathbf{x}_k) &+ \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \tfrac{L}{2} ||\mathbf{x} - \mathbf{x}_k||_2^2 + \delta_C(\mathbf{x}) \\ (I + \tfrac{1}{L} \partial \delta_C)(\mathbf{x}) &= \mathbf{x}_k - \tfrac{1}{L} \nabla f(\mathbf{x}_k) \\ \mathbf{x} &= \text{proj}_C \left(\mathbf{x}_k - \tfrac{1}{L} \nabla f(\mathbf{x}_k) \right) \end{aligned} \tag{2.60}$$

The optimization step in (2.60) is called projected gradient descent.

Example 2.8. [S^n_{++} projection] We can apply the concept of projections, to project any symmetric square matrix $\mathbb{R}^{n\times n}$ onto S^n_{++}. Our optimization problem is

$$\operatorname*{proj}_{S^n_{++}}(\hat{D}) = \operatorname*{arg\,min}_{D\in S^n_{++}} \left|\left|D - \hat{D}\right|\right|_F^2 \tag{2.61}$$

for any $\hat{D} \in \mathbb{R}^{n\times n}$. Recall the eigenvalue decomposition $D = U\Sigma U^\top$ and $\hat{D} = V\hat{\Sigma}V^\top$ and associated properties presented in Section 2.3, where $\Sigma \in \mathbb{R}^{n\times n}$ since $D \in S^n_{++}$.

$$\begin{aligned}
&\left|\left|D - \hat{D}\right|\right|_F^2 = \left|\left|U\Sigma U^\top - \hat{D}\right|\right|_F^2 \\
&= \left|\left|\Sigma - U^\top \hat{D} U\right|\right|_F^2 = \left|\left|\Sigma - \underbrace{U^\top V\hat{\Sigma}V^\top U}_{A}\right|\right|_F^2 \\
&= \textstyle\sum_i (\sigma_i - A_{ii})^2 + \sum_{i\neq j} A_{ij}^2
\end{aligned}$$

Note that the minimum is achieved if the second term of the sum is zero, which holds true if $U^\top V = \mathbf{I}$. Since we know that $D \in S^n_{++}$ if $\sigma_i \geq 0$ for $i \in [\![n]\!]$ (see (2.8)), the projection can be written as:

$$\operatorname*{proj}_{S^n_{++}}(\hat{D}) = V \operatorname{diag}(\max\{\hat{\sigma}_1, \varepsilon\}, \ldots, \max\{\hat{\sigma}_n, \varepsilon\})V^\top, \tag{2.62}$$

for $\varepsilon > 0$. This projection will later be used in Section 4.5 to ensure that our tensor field resulting from the optimization is still a valid vector space. ♦

2.10.3 Inverse Problems

Many optimization and especially inverse problems are not well-posed. Well-posed problems according to Hadamard [64], have the properties that

- A solution exists.
- The solution is unique.
- The solution continuously depends on the input parameters.

Problems that are not well-posed, i.e. they do not fullfil all of the three mentioned properties, are called ill-posed. Solving ill-posed problems can be a challenging task, as the latter two properties (non-uniqueness and sensitivity to noise) can yield highly different results for different initializations and levels of noise. A common approach to overcome this difficulty is the variational formulation of optimization problem, where an additional term — often called the regularizing function $\mathcal{R}$ — is employed as prior knowledge to discard unlikely solutions and smooth the minimization functional. Specifically, in many considered variational problems, the objective function f will take the following form

$$f(\mathbf{x}, \mathbf{b}) := \mathcal{D}(\mathbf{x}, \mathbf{b}) + \lambda\mathcal{R}(\mathbf{x}), \tag{2.63}$$

where we refer to $\mathcal{D}$ as the data term, computing the mismatch between the solution and given data and $\mathcal{R}$ as the regularizing function, imposing prior knowledge on the solution.

The proper choice of $\mathcal{R}$ has been a long debated topic in multiple fields for inverse problems. Throughout this work, we used an arguably very popular choice that has been successfully used in many inverse problems, such as image denoising: The Huber regularization [107], a smoothed Total Variation (TV) function (in its discrete form $\mathcal{R}(u) = \sum_i ||(\nabla_{\mathbf{x}} u)_i||_{H,\alpha}$), where $||\cdot||_{H,\alpha}$ is given by:

$$H_\alpha(\mathbf{x}) = ||\mathbf{x}||_{H,\alpha} = \begin{cases} \frac{1}{2\alpha}\,||\mathbf{x}||^2 & \text{if } ||\mathbf{x}|| < \alpha \\ ||\mathbf{x}|| - \frac{1}{2}\alpha & \text{else} \end{cases} \tag{2.64}$$

A plot of the Huber-norm can be seen in Figure 2.12.

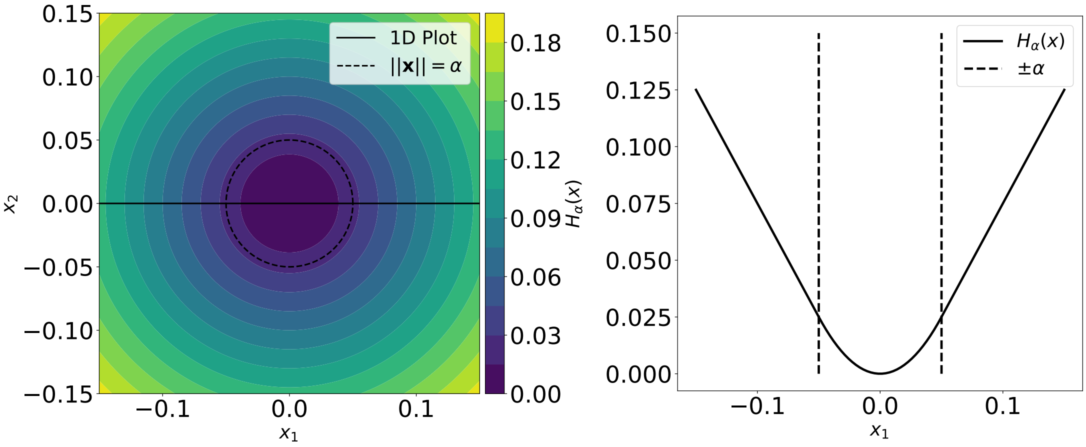


Figure 2.12: Visualization of the Huber norm of (2.64) for 2 and 1 dimensions (left and right) respectively for $\alpha = 0.05$.

Note that for this particular choice of H, it holds that $||\cdot||_{H,0} = ||\cdot||_2$.

Example 2.9. [Proximal Mapping of $H_\alpha(\cdot)$] To compute the proximal mapping

$$\operatorname*{prox}_{\tau H_\alpha(\cdot)}(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}} \frac{1}{2}\,||\mathbf{x} - \tilde{\mathbf{x}}||_2^2 + \tau H_\alpha(\mathbf{x}),$$

we distinguish the optimality condition of the two cases again:

$$\begin{aligned} ||\mathbf{x}|| < \alpha : \quad 0 &= \mathbf{x} - \tilde{\mathbf{x}} + \tfrac{\tau}{\alpha} \\ \mathbf{x} &= \tfrac{\tilde{\mathbf{x}}}{1+\frac{\tau}{\alpha}} \Rightarrow \left|\left|\tilde{\mathbf{x}}/\left(1+\tfrac{\tau}{\alpha}\right)\right|\right| < \alpha \\ ||\mathbf{x}|| \geq \alpha : \quad 0 &\in \mathbf{x} - \tilde{\mathbf{x}} + \tau \partial H_\alpha(\mathbf{x}) \\ 0 &\in \mathbf{x} - \tilde{\mathbf{x}} + \tau \partial \, ||\mathbf{x}||_2 \end{aligned}$$

To compute the proximal mapping of the Euclidean norm, we use the Moreau decomposition from (2.58). We know that $||\cdot||_2 = ||\cdot||_{2,*}$ and the proximal mapping of the convex conjugate of a norm is its projection onto the 1-ball of its dual norm (see Definition 2.2 and Lemma 1). This means for the convex conjugate Euclidean norm

$$\operatorname*{prox}_{\tau||\cdot||_*}(\mathbf{x}) = \operatorname*{proj}_{||\cdot||\leq 1}(\mathbf{x}) = \frac{\mathbf{x}}{\max(1, ||\mathbf{x}||)},$$

and thus

$$\operatorname*{prox}_{\tau||\cdot||}(\mathbf{x}) = \mathbf{x} - \tau \operatorname*{proj}_{\frac{1}{\tau}||\cdot||\leq 1}(\mathbf{x}/\tau) = \mathbf{x} - \tau \frac{\mathbf{x}/\tau}{\max(1, ||\mathbf{x}|| / \tau)} = \left(1 - \frac{\tau}{\max(\tau, ||\mathbf{x}||)}\right)\mathbf{x}.$$

So the resulting proximal mapping of the Huber norm is

$$\operatorname*{prox}_{\tau H_\alpha(\cdot)}(\mathbf{x}) = \begin{cases} \widehat{\mathbf{x}} & \text{if } ||\widehat{\mathbf{x}}|| < \alpha \\ \mathbf{x} - \frac{\tau \mathbf{x}}{\max(\tau, ||\mathbf{x}||)} & \text{else} \end{cases} \tag{2.65}$$

for $\widehat{\mathbf{x}} = \mathbf{x}/(1+\frac{\tau}{\alpha})$. The proximal mapping of the first component $\left(\operatorname{prox}_{\tau H_\alpha}(\mathbf{x})\right)_1$ is visualized in Figure 2.13. ♦

Such norms come in handy as regularizing functions for our inverse problems in variational formulation. A typical problem may consist of a least squares data term together with a regularization that keeps the parameters' norm low:

$$\min_{\mathbf{x}} \underbrace{\left|\left|\tilde{f}(\mathbf{x}) - \mathbf{b}\right|\right|_2^2}_{\mathcal{D}(\mathbf{x},\mathbf{b})} + \lambda \underbrace{||\mathbf{x}||_1}_{\mathcal{R}(\mathbf{x})}, \tag{2.66}$$

In case $\tilde{f}$ is a linear function, we refer to the class of problems in (2.66) as Least Absolute Shrinkage and Selection Operator (LASSO) problems [117]. Such class of problems are often optimized using Iterative Shrinkage and Thresholding Algorithm (ISTA) algorithms [12]. These optimization algorithms assume that there is a smooth part of the minimization problem f ($\mathcal{D}$ in this case), and a non-smooth part g ($\mathcal{R}$ in this case), for which the proximal mapping $\operatorname{prox}_{\lambda g}$ can be easily calculated. Using the optimality condition for (2.66) and quadratically bounding f using the descent lemma in (2.51), we can

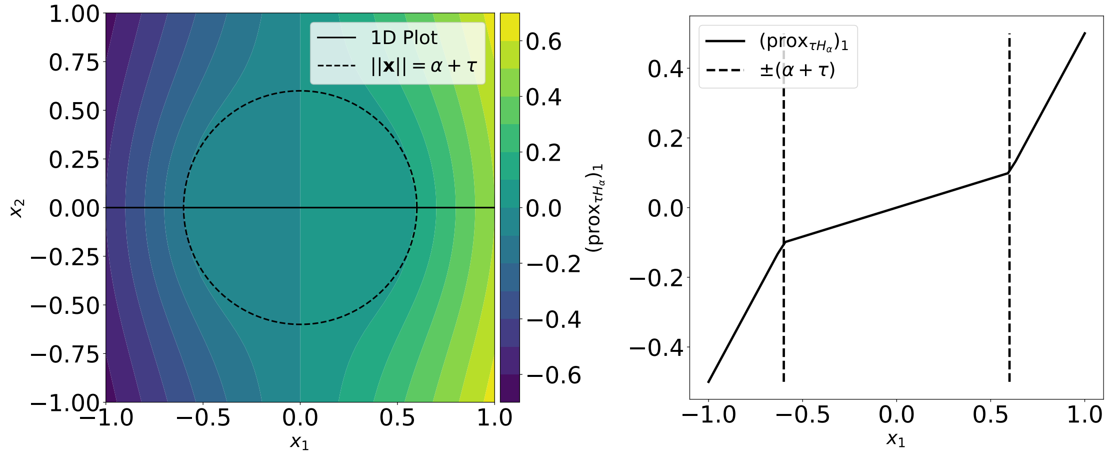


Figure 2.13: Visualization of the proximal mapping in (2.65) of the Huber norm for 2 and 1 dimensions (left and right) respectively. For the visualization we chose $\alpha = 0.1$ and $\tau = 0.5$ and show only the first component on the right. Note the similarity with the previously mentioned softshrinkage in (2.55), with the limit case $\alpha \to 0$ giving the same proximal mapping in 1D.

rewrite and solve the optimization problem as:

$$\begin{aligned} \arg\min_{\mathbf{x}} f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L}{2} ||\mathbf{x} - \mathbf{y}||_2^2 + \lambda g(\mathbf{x}) \\ 0 \in \nabla f(\mathbf{y}) + L(\mathbf{x} - \mathbf{y}) + \lambda \partial g(\mathbf{x}) \\ \mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \in \left(\mathbf{I} + \frac{\lambda}{L} \partial \right) \mathbf{x} \\ \mathbf{x} = \operatorname*{prox}_{\frac{\lambda}{L} g} \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \end{aligned} \tag{2.67}$$

In [12], the authors additionally introduced an over-relaxation step to increase the speed of such ISTA algorithms, which they called Fast Iterative Shrinkage and Thresholding Algorithm (FISTA). We will later use the Huber norm and its proximal mapping in Sections 4.5 and 4.6.2 for similar scenarios, but with non-linear functions $\tilde{f}$.

A problem yet not addressed though is the use of linear operators $K \in \mathbb{R}^n \times \mathbb{R}^n$ in g, i.e. problems of the type

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(K\mathbf{x}).$$

One such subclass of these problems is the TV-L2 denoising [107] problem

$$\min \frac{1}{2} ||\mathbf{x} - \widehat{\mathbf{x}}||_2^2 + \lambda \sum_i ||(\nabla \mathbf{x})_i||_2 . \tag{2.68}$$

Here, we want to minimize the mismatch between $\mathbf{x}$ and $\widehat{\mathbf{x}}$, while at the same time keep the solution $\mathbf{x}$ smooth across the domain. The TV prior is very famous in many applications beyond imaging and we have shown in our works [59–61] that it can be effectively used in our inverse PDE problems to discard solutions with too much variation. In particular, we show in [61] different TV weights (λ in (2.68)) influence the solution of the ill-posed problem, with higher values resulting in smoother conduction velocity tensors across the domain.

To solve such class of problems, we note that for convex functions, we can rewrite the problem using the convex conjugate g^* and Lagrangian multipliers $\mathbf{p}$ into a convex-concave saddle point problem:

$$\begin{aligned}
&\min_{\mathbf{x}} g(\mathbf{q}) + f(\mathbf{x}) \quad \text{s.t.:} \mathbf{q} = K\mathbf{x} \\
\Leftrightarrow &\min_{\mathbf{x},\mathbf{q}} \max_{\mathbf{p}} g(\mathbf{q}) + f(\mathbf{x}) + \langle K\mathbf{x} - \mathbf{q}, \mathbf{p} \rangle \\
\Leftrightarrow &\min_{\mathbf{x},\mathbf{q}} \max_{\mathbf{p}} g(\mathbf{q}) + f(\mathbf{x}) + \langle K\mathbf{x}, \mathbf{p} \rangle - \langle \mathbf{q}, \mathbf{p} \rangle \\
\Leftrightarrow &\min_{\mathbf{x}} \max_{\mathbf{p}} \underbrace{\left(- \max_{\mathbf{q}} -g(\mathbf{q}) + \langle \mathbf{q}, \mathbf{p} \rangle \right)}_{-g^*(\mathbf{p})} + f(\mathbf{x}) + \langle K\mathbf{x}, \mathbf{p} \rangle \\
\Leftrightarrow &\min_{\mathbf{x}} \max_{\mathbf{p}} f(\mathbf{x}) + \langle K\mathbf{x}, \mathbf{p} \rangle - g^*(\mathbf{p}),
\end{aligned} \tag{2.69}$$

where we use the definition of the convex conjugate from Definition 2.23. An efficient method to solve (2.69) is then given by the famous Primal-Dual algorithm [25]:

$$\left\{ \begin{aligned}
\mathbf{x}^{i+1} &= \operatorname{prox}_{\tau f}(\mathbf{x}^i - \tau K^* \mathbf{p}^i) \\
\bar{\mathbf{x}}^{i+1} &= \mathbf{x}^{i+1} + \theta \left(\mathbf{x}^{i+1} - \mathbf{x}^i \right) \\
\mathbf{p}^{i+1} &= \operatorname{prox}_{\sigma g^*} \left(\mathbf{p}^k + \sigma K \bar{\mathbf{x}}^{i+1} \right)
\end{aligned} \right. \tag{2.70}$$

for appropriately chosen parameters τ, σ, θ, such that $\tau\sigma\, ||K||_2^2 \leq 1$ and $\theta \in [0, 1]$.

The final concept of this section is closely linked to the proximal mapping in Definition 2.22 [13, p. 163]:

Definition 2.24. *Given a closed convex function $f : U \to (-\infty, \infty]$ and $\mu > 0$, the Moreau envelope of f is given by*

$$M_f^\mu(\mathbf{x}) = \min_{\mathbf{y} \in U} f(\mathbf{y}) + \frac{1}{2\mu} ||\mathbf{x} - \mathbf{y}||^2 \tag{2.71}$$

M is a smooth variant of the possibly non-smooth function f, where the smoothness can be varied by adjusting μ. For a known proximal mapping of f, M can be directly formulated as

$$M_f^\mu(\mathbf{x}) = f(\underset{\mu f}{\operatorname{prox}}(\mathbf{x})) + \frac{1}{2\mu}\left|\left|\mathbf{x} - \underset{\mu f}{\operatorname{prox}}(\mathbf{x})\right|\right|^2, \tag{2.72}$$

but more importantly, its gradient is given by:

$$\nabla_{\mathbf{x}} M_f^\mu(\mathbf{x}) = \frac{1}{\mu}(\mathbf{x} - \underset{\mu f}{\operatorname{prox}}(\mathbf{x})). \tag{2.73}$$

This way we can optimize a smoothed version of f if its proximal mapping is known.

Example 2.10. [Constraint Smoothing] This example is a regularized variant of the problem presented in [26, p. 29 ff]. Consider for $K \in \mathbb{R}^{n\times m}, A \in \mathbb{R}^{k\times m}, \mathbf{y} \in \mathbb{R}^m, \mathbf{b} \in \mathbb{R}^n$, and a convex set C, the following optimization problem:

$$\min_{\mathbf{y}} f(\mathbf{y}) = \min_{\mathbf{y}} \frac{1}{2}\left|\left|K\mathbf{y} - \mathbf{b}\right|\right|_2^2 + \frac{\gamma}{2}\left|\left|A\mathbf{y}\right|\right|_2^2 + \delta_C(\mathbf{y}),$$

which is a convex problem that could be solved e.g. using projected gradient descent. An alternative approach consists in smoothing the problem by means of the Moreau envelope:

$$M_f^\tau(\bar{\mathbf{y}}) := \min_{\mathbf{y}} \frac{1}{2}\left|\left|K\mathbf{y} - \mathbf{b}\right|\right|_2^2 + \frac{\gamma}{2}\left|\left|A\mathbf{y}\right|\right|_2^2 + \delta_C(\mathbf{y}) + \frac{1}{2}\left|\left|\mathbf{y} - \bar{\mathbf{y}}\right|\right|_B^2,$$

for $B \in S_{++}^m$. By choosing $B = \frac{1}{\tau}\mathbf{I} - K^\top K - \gamma A^\top A$, we can explicitly compute $M_f^\tau(\bar{\mathbf{y}}) = f(\hat{\mathbf{y}})$ with

$$\hat{\mathbf{y}} = \underset{\tau C}{\operatorname{proj}}\left(\bar{\mathbf{y}} - \tau K^\top(K\bar{\mathbf{y}} - \mathbf{b}) - \tau\gamma A^\top A\bar{\mathbf{y}}\right).$$

Note that $B \in S_{++}^m$ requires that $\tau < \left(\left|\left|K^\top K\right|\right| + \gamma\left|\left|A^\top A\right|\right|\right)^{-1}$, where the exact value again influences the smoothness of the Moreau envelope, similar to μ in (2.72). The gradient of the Moreau envelope M_f^τ is given by

$$\nabla M_f^\tau(\mathbf{y}) = \frac{1}{\tau}(\mathbf{y} - \hat{\mathbf{y}}), \tag{2.74}$$

which can be readily used in an unconstrained optimization algorithm like Limited memory Broyden Fletcher Goldfarb Shanno (L-BFGS) [21]. ⧫

Note that both in Sections 4.5 and 4.7 we will employ the Moreau envelope of Example 2.10 to use the very efficient L-BFGS algorithm to optimize constrained inverse problems.

3 Physiological Foundations

If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is. (John v. Neumann)

Contents

3.1 From chemical elements to electrical activation 47
3.2 Anatomy & ECG 54
3.3 Lead Fields 57

This section will take the mathematical and more general definitions introduced in Chapter 2 and use it to define the physiological foundations of how electrical signals in the heart are generated and propagated. In Section 3.1, we will show how microscopic phenomena and properties at cell level, lead to the electrical excitation of cells and how this behavior can be mathematically modelled. We will show how these models work in each individual cell, but also how they jointly make up the activation patterns on a spatial level. In all of our previous works, we readily used the eikonal equation as a compromise between model fidelity and computational efficiency. Here, we also show how from chemical processes at the cell levels and their interactions with each other, we arrive in the end at the eikonal approximation. This is based on and is only a short summary of all processes described in [78] and [51] among others.

Section 3.2 will give some insights on the actual ECG setup, necessary to better understand how the electrical activation of the heart is captured using ECGs.

Section 3.3 will conclude this chapter with the underlying physics of the ECG procedure and how it can be replicated in-silico using lead field theory.

3.1 From chemical elements to electrical activation

It has been suspected for a long time that the cell membrane separated the different ionic concentrations, but the breakthrough was finally achieved in 1952 by Hodgkin and Huxley, when new measurement techniques (the space clamp technique) allowed to measure the transmembrane voltage ex-vivo for the very first time in a single giant squid axon [70]. This led to a series of fives articles and earned Hodgkin and Huxley the nobel prize. Already in this series of articles, they formulated an equivalent electrical circuit to describe the electrical behavior of a cell over the duration of an action potential. We call this model from today's point-of-view the Hodgkin-Huxley model, predecessor to many subsequently developed models up until this day [86].

The two main driving forces between intra- and extracellular space are the chemical gradient (according to Fick's first law) and the electrical gradient. If both of these forces have the same magnitude in the opposite direction, the flow of chemicals will stop and the concentrations will enter a steady or equilibrium state. The associated measurable electrical gradient in this equilibrium state is called the equilibrium, reversal or Nernst potential. In cells in particular, the cell wall is only permeable to positive ions as shown in Figure 3.1. An important property of cardiac cells in particular is their excitability: In the resting state, the membrane remains at its so-called resting potential until an external stimulus current is applied. If this current exceeds a certain threshold and is applied long enough, the cell will initiate its characteristic action potential cycle during. This process is outlined in Figure 3.2. During the process of an action potential, the flux of chemicals will rapidly change between channels before returning to the resting state. This is modelled using one or multiple gating variables per ion channel.

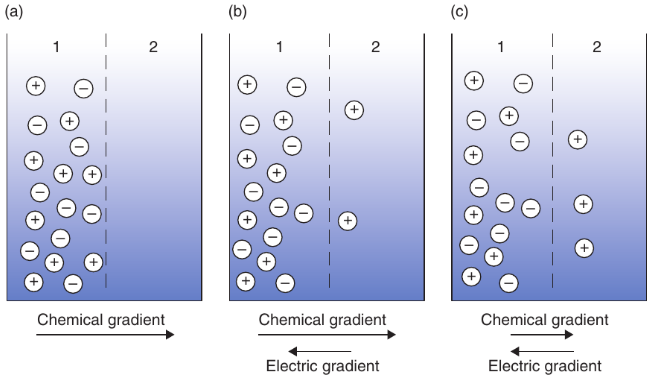


Figure 3.1: Consider a vessel divided by a membrane into two compartments, only permeable to positive ions. From an initial ionizable solution in compartment 1 (a), the positive ions will diffuse into compartment 2 through the chemical gradient and through the separation of positive and negative ions, induce an electrical gradient (b). This flow will stop once we reach the equilibrium state where the chemical and electrical gradient have the same magnitude in opposite directions (c). Figure from [71].

The circuit diagram of the Hodgkin–Huxley model can be seen in Figure 3.3, where the driving ionic currents of the cell identified were potassium (K), sodium (Na) and calcium (Ca). With each conductance (e.g. g_K), there is a gating variable associated that mimics the ionic channel behavior. The exact equations and values for each of these gating variables are not summarized here, but can be found in greater detail in [78] and [51].

Remark 3.1. *The important property (from an evolutionary point of view) of this electrical propagation through the heart is not of electrophysiological, but rather mechanical*

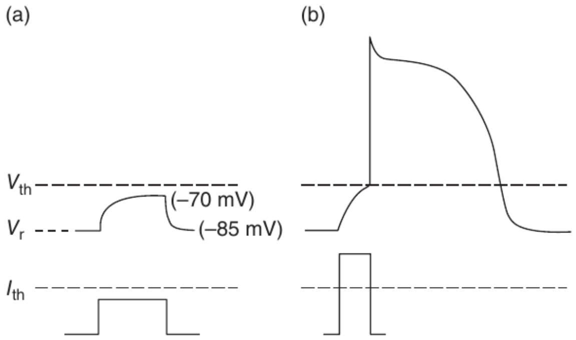


Figure 3.2: Cardiac cells are excitable, meaning that an external stimulus current can initiate an action potential cycle. If the external stimulus (bottom) is only shortly applied, or does not exceed a needed threshold, the transmembrane voltage will quickly return to its resting potential (left). In contrast, if we exceed the needed thresholds, the cell will enter the characteristic action potential cycle in which the voltage will remain high, even if the external stimulus is removed. Figure from [71].

nature: The electrical activation forces a contraction of the heart muscle to pump blood throughout the body. This is achieved by the filament structure and the Ca concentration in the cells. As outlined in Table 3.1, the resting state intracellular Ca-concentration is very low, but will rapidly increase as the tissue is excited. The result of this high intracellular Ca concentration causes the so-called myofilament structure to pull on the thin filaments [78, chapter 18]. This causes the muscles to contract, before returning to the resting state with low intracellular Ca-concentration, relaxing the muscle.

While the Hodgkin-Huxley model is already an efficient, yet accurate simplification of the real underlying chemical processes, further simplifications can be made. Successive research introduced simpler models that approximate the same emergent behavior, using only one collective gating variable h and inward/outward current [86]. The constituting relation can be expressed as

$$\frac{\mathrm{d}V_m}{\mathrm{d}t} = I_{in}(V_m, h) + I_{out}(V_m) + I_{stim}(t), \tag{3.1}$$

where V_m, I_{in}, I_{out} and I_{stim} are the transmembrane voltage, intracellular, extracellular and stimulus current respectively, defined as:

$$\begin{aligned}
C(V_m) &= V_m^2(1-V_m) \\
I_{in}(V_m,h) &= \frac{hC(V_m)}{\tau_{in}} \\
I_{out}(V_m) &= -\frac{V_m}{\tau_{out}} \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \begin{cases} \frac{1-h}{\tau_{open}} & \text{if } V_m < V_{gate} \\ \frac{-h}{\tau_{close}} & \text{else} \end{cases}
\end{aligned}$$

The constants $\tau_{in/out/open/close}$ and V_{gate} control the overall shape of the resulting action potential. By choosing $t \in [0, 1000]ms, \tau_{close} = 150, \tau_{open} = 120, \tau_{out} = 6, \tau_{in} = 0.3$ and $V_{gate} = 0.13$ and injecting a small I_{stim}, we can generate the characteristic action potential of a single cell with an Action Potential Duration (APD) of ≈ 200ms, visualized in Figure 3.4. Note that here our action potential is rescaled to $V_m \in [K_0, K_1]$ by using $\bar{V}_m = K_0 + V_m(K_1 - K_0)$ with the chosen resting and plateau potential $K_0 = -85mV$ and $K_1 = 30mV$ respectively, as already proposed in [86]. This model can be very useful in cases where we are only interested in the transmembrane voltage and in-/outward currents, but it can not be used to analyze individual ionic currents.

Element	Extracellular concentration [mM]	Intracellular concentration [mM]	Nernst potential [mV]
Na^+	145	15	60
Cl^-	100	5	-80
K^+	4.5	160	-95
Ca^{2+}	1.8	10^{-4}	130

Table 3.1: Intra- and extracellular concentration and Nernst potential values for ventricular myocytes in the resting state. Values from [51, p. 25]

Up until this point, we considered only single cells and their membrane voltage to be completely isolated from the outside world except for the stimulus current. In reality however, the emergent behavior of the electrical activation of the heart is a consequence of its resistive interconnections, referred to as gap junctions [78], and networks that propagate the electrical activation in an ordered fashion between cells. A straight-forward extension to our exemplary single-cell electrical circuit in Figure 3.3, is to consider a chain/cable of cells, such as is visualized in Figure 3.5. Each chain is a small differential element of the single-cell electrical circuit of Figure 3.3. In this scenario, the so-called axial currents I_e and I_i propagate along the cable, introducing space and thus replace our initially considered ODE with a PDE. By a clever change of variables, we can reformulate the cable equation as the 1D-PDE

$$\frac{V_\mathrm{m}}{\partial t} = \frac{\partial^2 V_\mathrm{m}}{\partial x^2} + f(V_\mathrm{m}, t), \tag{3.2}$$

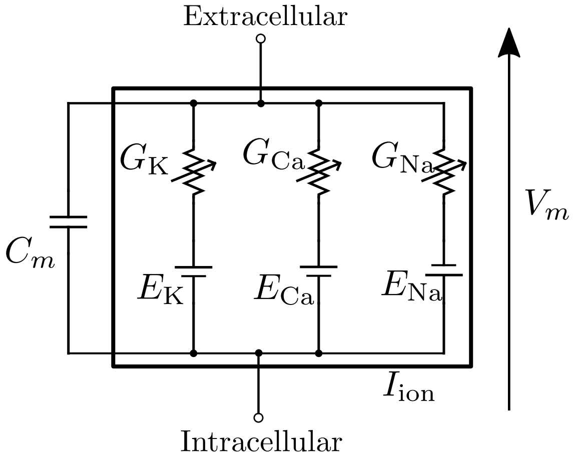


Figure 3.3: Electrical circuit diagram of a single cell membrane, similar to the one presented in [70]: The transmembrane voltage V_m is generated by Nernst potentials, responsible for a separation of chemical elements leading to the characteristic action potential inside a cell. The identified main drivers of the resulting action potential were potassium (K), calcium (Ca) and sodium (Na). Though more complex models have been developed [84], this model remains a good approximation with only three ion species involved.

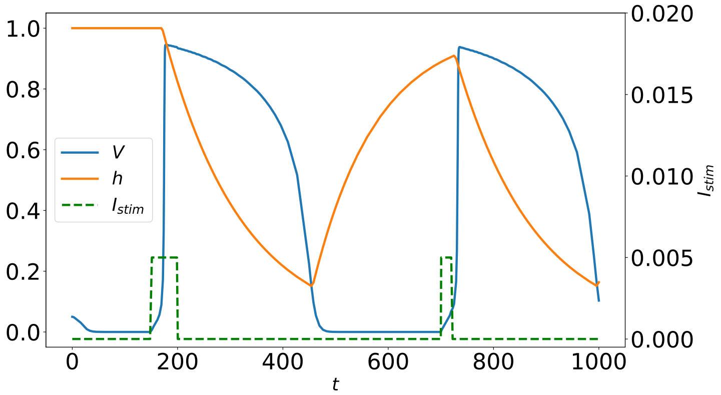


Figure 3.4: Approximation of the transmembrane action potential using a simplified state model with only one gating variable [86].

for some f. For more information on how this is done, we refer to [78, p. 251 ff].

When considering the behavior of multiple interconnected cells through (3.2) for a cable containing only homogeneous cells, it is useful to define traveling waves. Traveling waves here denote propagation in excitable systems that travel at constant velocity with a fixed shape and are categorized into traveling fronts that shift from one to another stable state, or traveling pulses, which will return to the original state after a fixed amount

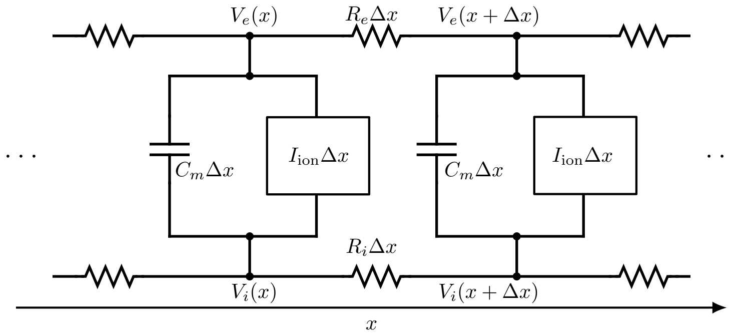


Figure 3.5: When considering a 1D compound of ionic electrical circuits, visualized in Figure 3.3, we can consider discrete differential chain of these components. As $\lim_{\Delta x \to 0}$, we end up with (3.2).

of time. Here, we only consider traveling fronts, which are bistable excitations, moving smoothly between two resting states:

$$\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2} + f(V) \tag{3.3}$$

where f is zero at three different positions between 0 and 1, i.e. $f(0) = f(\alpha) = f(1) = 0$ for $0 < \alpha < 1$. We can see that (3.3) is a special case of the cable equation in (3.2). A special class of solutions — that we are mostly interested in — take the form

$$V(x,t) = U(x+ct) = U(\xi) \tag{3.4}$$

with a physiological, currently unknown constant speed c depending on $f(V)$. This function assumes a fixed waveform U that moves with constant speed c across the domain and is simply shifted over time, but retains its shape. Reintroducing this waveform into the original bistable equation (3.3), we end up with:

$$U_{\xi\xi} - cU_\xi + f(U) = 0. \tag{3.5}$$

This reduces our PDE in (3.3) again to an ODE. We only require that $\lim_{\xi \to \pm\infty} f(U(\xi)) = 0$.

Consider now a specific function $f(U) = -d_1^2(U-U_0)(U-U_1)(U-U_2)$ with $U_0 < U_1 < U_2$. When guessing that $W = -d_2(U-U_0)(U-U_2)$ and substituting into (3.5), we see that it must hold that

$$d_2^2(2U - U_0 - U_2) - cd_2 - d_1^2(U-U_1) = 0.$$

By additionally assuming $d_2 = d_1/\sqrt{2}$ and $c = \frac{d_1}{\sqrt{2}}(U_2 - 2U_1 + U_0)$, we conclude [78]

$$U(\xi) = \frac{U_0 + U_2}{2} + \frac{U_2 - U_0}{2} \tanh\left(\frac{d_1}{\sqrt{2}} \frac{U_2 - U_0}{2}\right) \tag{3.6}$$

This goes to show that assuming reasonable analytical functions, we can end up with very nice closed-form solutions for traveling waves. The closed solution of the waveform in (3.6) can be later seen in Section 4.7, Figure 4.23. We will also use (3.6) in the mentioned section to compute the transmembrane voltages on the domain Ω from an eikonal solution.

Extending the presented 1D theory to multiple dimensions is straight-forward for planar waves traveling only in one direction $\mathbf{n}$, where (3.4) becomes

$$u(\mathbf{x}, t) = U\left(\frac{\langle \mathbf{n}, \mathbf{x}\rangle - c(\mathbf{n})t}{g(\mathbf{n})}\right), \tag{3.7}$$

where $c(\mathbf{n})$ and $g(\mathbf{n})$ are the directionally dependant speed and space constant, respectively. In contrast, for waves with curvature, we can extend (3.2) to

$$\frac{\partial u}{\partial t} = \operatorname{div}(D\nabla u) + kf(u), \tag{3.8}$$

for the conduction velocity tensor $D \in S^n_{++}$ and a time constant k. The problem in (3.8) is hard to study, which is why we now instead consider a moving coordinate system $x = X(\xi, t)$, such that the main component ξ_1 is normal to the level surface of u at time t. Consider now a level set function

$$S(\mathbf{x}, t) = \begin{cases} > 0 & \text{if wave has already passed } \mathbf{x} \\ < 0 & \text{else} \end{cases} \tag{3.9}$$

With the help of this level set function, it is shown in [78] that $0 = \langle \nabla S, X_t\rangle + S_t$ must hold and hence we can rewrite (3.8) as

$$\frac{\partial S}{\partial t} = ||\nabla S||_D + \lambda\, ||\nabla S|| \operatorname{div}\left(\frac{\nabla S}{||\nabla S||}\right) \tag{3.10}$$

Neglecting the curvature, we end up with

$$\frac{\partial S}{\partial t} = ||\nabla S||_D \tag{3.11}$$

Since $S(\mathbf{x}, t)$ is a function of the level surface at time t, we can deduce that the velocity in normal direction $\mathbf{n}$ of the current surface $S(\mathbf{x}, t) = R_t(\mathbf{x})$ must fullfil

$$\langle R_t, \mathbf{n}\rangle_{D^{-1}} = 1 = ||\nabla\phi||_D = \sqrt{\langle D\nabla\phi, \nabla\phi\rangle}, \tag{3.12}$$

where ϕ is now our desired quantity of earliest arrival time. Equation (3.12) shows how from this level set definition, we arrive at the eikonal solution, which, instead of solving a PDE over time, only computes the solution to the non-linear PDE in (3.12). The approach to use an indicator function, following the motion of an interface, is commonly known as the level set method [93] and was used in [112, 118] to derive the well known Fast Marching Method (FMM). In practice, dropping the time as a variable makes the computations both efficient and allows for lower spatial resolution of the computational domain [33] (1mm in the context of the heart), which makes it especially useful for gradient-based inverse problems. This equation is heavily employed and relied upon throughout the whole thesis and will appear in all proposed methods in one form or another. Recall from Section 2.8.1 that in this work we only consider viscosity solutions of the problem in (3.12), where we mostly use the computationally efficient Fast Iterative Method (FIM) presented in [53].

Remark 3.2. *Note that the presented derivation on how to arrive at the eikonal equation follows the train of thought in [78, p. 300 ff], but there are multiple paths to arrive to this conclusion. Another popular way, not shown here, is to derive the bidomain equation from the ionic model, simplify it to the monodomain model and derive the eikonal equation from the monodomain model. We refrained from using the derivation over the monodomain model, since we only marginally used the bi- and monodomain equations throughout this work.*

3.2 Anatomy & ECG

From a macroscopic anatomical view, the heart is divided up into atria that receive the blood from the body and lungs, and ventricles that pump the blood through this system. The cardiovascular system can be seen in this context as an electrically driven, coordinated pump: Oxygen-deprived blood arrives at the RA and is subsequently pumped through the Right Ventricle (RV) over the pulmonary arteries to the lungs (more precisely the alveola). Oxygen-satured blood from the lungs in contrast is received from the pulmonary veins in the LA and pumped over the aorta from the left ventricle throughout the body. In this thesis however, we are more interested in the controlling electrical waves that synchronize the mechanical action, as already briefly outlined in Chapter 1: From a macroscopic view, the heart is activated from the sinoatrial node, located in the RA and first propagated through LA and RA, before being slightly delayed at the atrioventricular node. The delayed ventricular activation then starts from the atrioventricular node rapidly through the HPS system, where the right and left bundle activate their respective ventricles. The tissue itself propagates the electrical activation to the still unexcited tissue, but at a much slower pace. Figure 3.6 depicts both mentioned views on cardiac anatomy.

Note that the mentioned sequence of electrical, coordinated activations is what can be monitored on an ECG, which measures electrical activation on the skin, as a consequence of cardiac activation. The ECG is one of the favored tools for cardiologists to diagnose cardiovascular diseases, since common electrophysiological diseases, such as AF and Left

Bundle Branch Block (LBBB), can be directly observed with minimal cost and harm to the patient. One of the arguably most popular ECG measurement technique is the 12-lead ECG, commonly encountered in the clinics. In the setup of a 12-lead ECG, 9 electrodes are used to get 12 timeseries recordings by taking linear combinations between them: Three limb electrodes with potentials Φ_L, Φ_R, Φ_F for left-, right arm and foot respectively, are used together with six precordial leads $V1$ through $V6$. In order to get the 12-lead ECG recordings, we use the following equations (see [51] for more details):

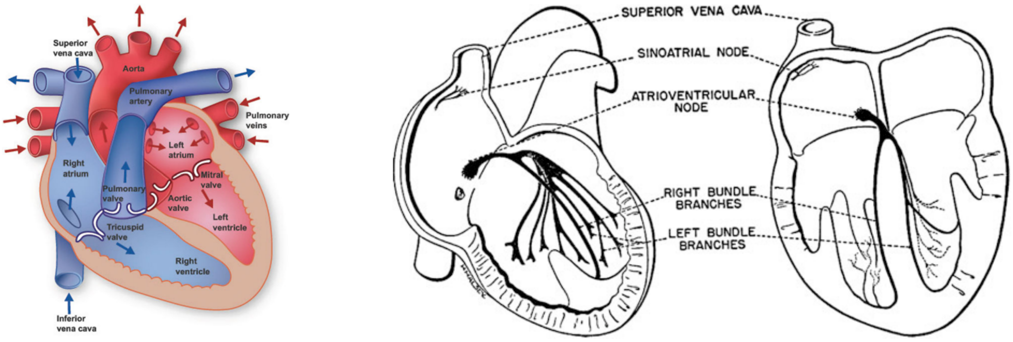


Figure 3.6: Anatomy of the heart from two different perspectives: Left shows the pathways that dictate the blood flow, while right shows the electrical pathways from the sinoatrial node throughout the heart. Figures from [51].

The Einthoven limb leads are defined as:

$$\begin{aligned} V_I &= \Phi_L - \Phi_R \\ V_{II} &= \Phi_F - \Phi_R \\ V_{III} &= \Phi_F - \Phi_L (= V_{II} - V_I \text{ by Kirchhoff's Law}) \end{aligned}$$

The augmented Goldberger leads are then given by:

$$\begin{aligned} V_{aV_L} &= \Phi_L - \frac{\Phi_F + \Phi_R}{2} \\ V_{aV_F} &= \Phi_F - \frac{\Phi_L + \Phi_R}{2} \\ V_{aV_R} &= \Phi_R - \frac{\Phi_L + \Phi_F}{2} \end{aligned}$$

Finally, for the precordial leads, we need to define the Wilson central terminal as:

$$\Phi_{WCT} = \frac{\Phi_R + \Phi_L + \Phi_F}{3}$$

With this, V_1 through V_6 are given by:

$$V_i = \Phi_{V_i} - \Phi_{WCT}$$

for $i \in [\![6]\!]$. In Figure 3.7, the exemplary placement of the 9 electrodes can be seen.

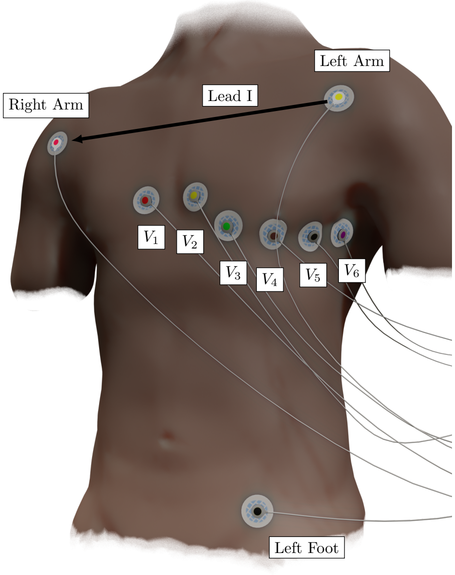


Figure 3.7: Here we show how the 9 electrodes of a 12-lead ECG are placed on the torso, and how to derive the exemplary Lead I from them. Note that the measured 12 leads are constructed as linear combinations of the 9 measurement electrodes, further detailed in the text.

Remark 3.3. *The leads V_1 through V_6 are also sometimes called unipolar leads, which can lead to the mislead thinking of them being compared to an external ground. While Φ_{WCT} remains mostly constant over the duration of the recording, it is important for the lead field theory, introduced in Section 3.3 that this point can be defined as a linear combination of other points present on the domain.*

The prominent activation observed on the ECG is usually seen to be divided into 5 parts: P, Q, R, S and T. The P-wave marks the atrial activation, while the QRS-complex is witnessed during the activation of Left Ventricle (LV) and RV. The T-wave is a consequence of the repolarization, i.e. the action potential drop-off after the APD, of the whole heart before the next cycle begins. Figure 3.8 outlines a schematic of an ECG, similar to how it is encountered in clinics on Lead I. There exist other forms of electrical recordings of cardiac activity, such as the Vector ECG or body potential surface maps, not further detailed in this work (for more information, see [51, 78]).

The introduced leads help clinicians in diagnosing disturbances of activation and repolarization patterns, as ECG leads record the time-dependant projection of dipoles within the heart onto the lead axes. The next section will introduce the link between electrical activity at the cellular level from Section 3.1 and current flow in the surrounding torso and associated potential differences, crucial for computing the ECG.

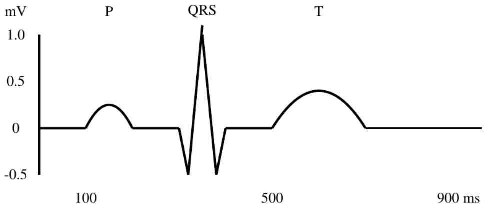


Figure 3.8: Schematic representation of the ECG from [116]

3.3 Lead Fields

In what follows, we shortly summarize the heart dipole theory from [116, p. 21 ff], [51] to set the necessary foundation and later end up with the lead field equation. The lead field is the main connection between the electrical activation inside each cell and the consequent potentials measured between multiple electrodes. All of the resulting lead field equations will then be readily used in Section 4.7 to compute the electrical activation observed on leads of a minimal torso model. Maxwell's equation will serve as a starting point for this purpose, where we refer to [46] for more details on the origin and nature of these equations.

Consider a torso domain Ω, with the heart $\Omega_H \subset \Omega$ inside the torso. We start with the general Maxwell equation for volume conductors, for an electrical and magnetic field E and B respectively:

$$\nabla \times E(\mathbf{x}) + \frac{\partial B(\mathbf{x})}{\partial t} = 0,$$

where $\times$ denotes the curl operator in this context. The torso is assumed to be quasi-constant, i.e. the change of the magnetic field is assumed to be negligible. From this it follows

$$\nabla \times E(\mathbf{x}) = 0.$$

We rewrite this equation in terms of the potentials $\phi(\mathbf{x})$:

$$E(\mathbf{x}) = -\nabla\phi(\mathbf{x})$$

The current inside such a conductor can be written as

$$I = GE = -G\nabla\phi$$

which, in the absence of sources, becomes

$$\nabla \cdot I = 0 = \nabla \cdot G \nabla \phi \tag{3.13}$$

A reasonable assumption is that the body is surrounded by air or other insulating media. We conclude from this that

$$\mathbf{n}(\mathbf{x}) \cdot I(\mathbf{x}) = 0 = \mathbf{n}(\mathbf{x}) \cdot G(\mathbf{x}) \nabla \phi(\mathbf{x})$$

for $\mathbf{x} \in \partial\Omega$ with $\mathbf{n}$ being the surface normal. The PDE torso model with no cardiac sources is thus given by

$$\begin{cases} \nabla \cdot (G \nabla \phi) = 0 & \text{in } \Omega \\ \mathbf{n} \cdot (G \nabla \phi) = 0 & \text{on } \partial\Omega \end{cases}. \tag{3.14}$$

This combined torso model is already includes a lot of the necessary foundations, but neglects the cardiac sources inside the heart as a consequence of chemical fluxes, as previously discussed in Section 3.1. To further extend this model recall from Section 3.1 that the cellular domain is separated into intra- and extracellular spaces [116, p. 26 ff]. The currents I_i and I_e of the two domains are then

$$\begin{cases} I_i & = -G_i \nabla \phi_i \quad \text{in } \Omega_i \\ I_e & = -G_e \nabla \phi_e \quad \text{in } \Omega_e, \end{cases} \tag{3.15}$$

where G_i, G_e are the conductivity tensors and ϕ_i, ϕ_e are the potentials for the intra- and extracellular space respectively. Further, we can define

$$\begin{cases} \nabla \cdot G_i \nabla \phi_i & = \beta I_{\text{ion}} + \frac{\partial q_i}{\partial t} \quad \text{in } \Omega_i \\ \nabla \cdot G_e \nabla \phi_e & = -\beta I_{\text{ion}} + \frac{\partial q_e}{\partial t} \quad \text{in } \Omega_e, \end{cases} \tag{3.16}$$

where q_i, q_e are the accumulations of charges in the intracellular and extracellular domain respectively, for which $\frac{\partial q_i}{\partial t} + \frac{\partial q_i}{\partial t} = 0$ holds. The voltage were are interested in (that defines the cardiac sources) is the transmembrane voltage, given by $V_{\text{m}} = \phi_i - \phi_e$. Combining and expressing (3.16) using V_{m}, we get

$$\nabla \cdot \underbrace{(G_i + G_e)}_{G} \nabla \phi_e = -\nabla \cdot G_i \nabla V_{\text{m}} \tag{3.17}$$

We can see now that (3.14) uses the so-called bulk conductivity given by $G = G_i + G_e$ and the observable extracellular potentials ϕ_e. The right term of (3.17) now describes the formerly missing cardiac sources.

While (3.17) is a useful formulation and allows us to compute potential field throughout the whole domain, it is sometimes far more useful to compute the potentials only at certain points using the “Integral Formulation”. Here, the measured potential difference between two points is given by

$$V(t) - V_{\mathrm{ref}} = \int_{\Omega} (-G_i(\mathbf{x}) \nabla V_{\mathrm{m}}(\mathbf{x}, t)) \cdot \nabla Z(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{3.18}$$

for a given lead field Z that describes the measured potential difference direction between electrodes placed on the torso, inside our domain. The remaining question is how to compute the lead field Z. As pointed out in [85], Z can be computed by simulating an unit current through the torso domain from the positive to the negative electrode during no activity of the heart:

> *This means that the current field, Z at a point in the heart, resulting from the introduction of a unit current into the lead, has the same direction and intensity as the Burger lead vector of that lead with respect to electromotive forces located at that point.* ([85])

Intuitively, we pose this observation as the PDE problem

$$\nabla \cdot (G \nabla Z) = \begin{cases} -1 & \text{at positive electrode} \\ 1 & \text{at negative electrode} \\ 0 & \text{ else} \end{cases}, \tag{3.19}$$

though it is important to note that this problem needs certain setups of the electrodes to avoid being ill-posed. Section 4.7.2.2 will shortly remark on this problem. Figure 3.9 shows how the computed lead field for lead I on a simple torso model with lungs and ribs using (3.19) looks like. The white streamlines between the electrodes are computed using ∇Z.

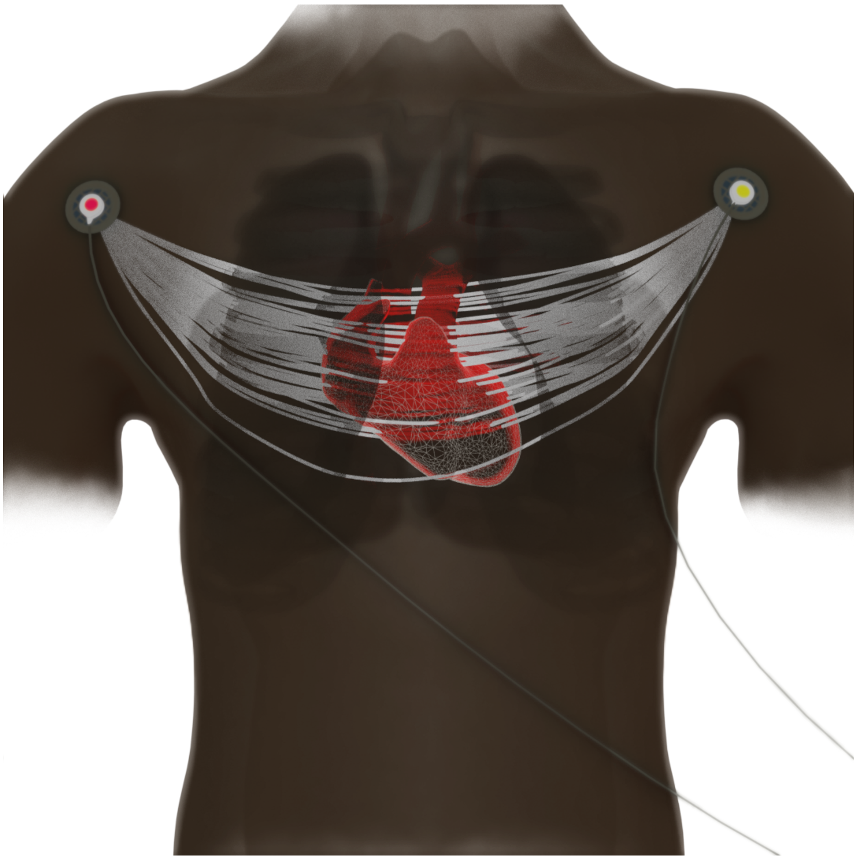


Figure 3.9: Visualization of the lead field for lead I from Figure 3.8, computed using (3.19) on a torso model. The visualized streamlines (white) show how the lead field ∇Z is orientated inside the heart (red) between two electrodes and helps us in understanding which electrical propagation ∇V_m will be visible by the lead. From (3.18), we can see that the ECG amplitude is maximized for propagation in the streamline direction, while being low for propagation normal to the streamlines (i.e. $\langle \nabla V_\mathrm{m}, \nabla Z \rangle = 0$).

4 Proposed Methods

Structures are the weapons of the mathematician. (Nicolas Bourbaki)

Contents

4.1 Related Methods 61
4.2 Methodological Foundations 64
4.3 Solving the Anisotropic Eikonal Equation - The Fast Iterative Method (FIM) 65
4.4 Runtime 69
4.5 FIMIN: A direct approach at solver-based gradient computation 70
4.6 Endocardial contact mapping 91
4.7 Identifying cardiac initiation sites through geodesics - GEASI 106

This chapter will introduce list and further explain all methods that were developed in the past few years. First, we will look at work of other research groups in Section 4.1, whereas the later chapters will present our proposed methods. Note that this chapter will concentrate on the actual method, since preliminaries, such as most of the needed equations and physiological foundations have already been introduced and explained in Chapters 2 and 3. Some concepts will be revisited as is seen fit to improve the flow of reading.

4.1 Related Methods

As already outlined in Chapter 3, the electrical propagation on a macroscopic level is anisotropic as a consequence from the fiber alignment inside the heart [32]. Early studies by Streeter et al.already revealed that the heart muscle has a complex, yet rather smooth fiber orientation in transmural direction [115]. Such a-priori knowledge was used in e.g. [9] to define a mathematical model that mimics this behavior and can be applied to arbitrary heart geometries acquired through e.g. Magnetic Resonance Imaging (MRI). Along the same lines, much scientific effort was focused on the generation of the Purkinje-network that dominates most of the cardiac activation in the ventricles [71]. Many procedures have been therefore developed to estimate the Purkinje network mostly from EAMs [7, 94, 119].

E.g. in [94], fractal laws were applied to generate Purkinje networks from these measurements. In contrast, in the classical Electrocardiographic Imaging (ECGi) problem, one seeks to infer an epicardial LAT map from the ECG [48, 67, 72, 120]. Modern approaches extended this idea to also infer e.g. transmembrane voltages [68, 124]. This poses several challenges, as in usual clinical setups, the electrical potential is only measured at very few points. To overcome this hurdle, special measurement technology, the Body Potential Surface Map (BPSM) vest has been developed, consisting of a matrix array of measurement nodes distributed over the whole torso. This offers a lot of spatial information on the electrical activation on the whole torso, potentially alleviating the ill-posedness of the inverse procedures [63]. The transfer from the epicardial to the torso activation is assumed to be linear and can be therefore efficiently modelled. However, the problem is still ill-posed since even the resolution of the BPSM can not match the required the resolution at the epicardium. Additionally, the measurements are also corrupted by noise, requiring additional regularization to obtain reasonable results. But even if one is to solve the ECGi problem perfectly, the epicardial activation gives only a glimpse on the total activation inside the heart, since much of the complexity already appears in the endocardial and transmural activation (see Section 3.2).

One additional problem that needs to be solved are the model assumptions and the associated computational costs. High-fidelity models already pose a difficult optimization problem just to model the forward problem, i.e. the problem of computing activation maps, ECGs or similar, given the parameters. To achieve a e.g. least squares fit to given data on top (compare Section 2.10), the model has to be solved several times and its gradient have to be computed for many of the considered optimization methods. Several possibilities have been developed to tackle the problem of complex forward models: Surrogate models can be used to replace the original difficult problem with a much simpler, yet related problem [41]. Alternatively, this simplification can also be posed in form of a reduced order model [126]. In contrast, many works concentrate on modeling and personalizing only certain parts of the activation like the Purkinje-network, see e.g. the before-mentioned work [94].

The parameterization and subsequent optimization of the considered models can be achieved in several ways: Fiber and front velocities/angles can be locally estimated from LAT measurements, acquired through contact recording. The two main drawbacks of these methods is the necessity of having contact recordings and the susceptibility to noise. Noisy measurements either need to be discarded by finely tuned heuristics or manually, as there is no inherent global consistency mechanism that could help in identifying noisy measurements. Anisotropic conductivity in these techniques can be deduced from front velocity and prior knowledge on fiber structure (rule-based or atlas-based), or by combining multiple activation maps [103]. In contrast, one can also use the assumed PDE model and enforce the model assumptions on the measurements. This can either be enforced pointwise, yielding for instance PDE-constrained optimization [8], or act as a penalization term [109]

Like in many other fields, also ML methods can be investigated. Despite popular belief, the roots of ML inspired-techniques, especially involving Neural-networks, reach back to the 60s [105]. In recent years however, they managed to revolutionize many fields like computer vision, dynamic systems and robotics [74]. This shift in paradigm was not only possible because the methods improved, but lends a big portion of its success to the increased computational power which was given to researchers around the globe by highly multi parallel computational devices at relatively low cost, like Graphics Processing Units (GPUs) (further outlined in Section 4.4). This change has already affected the field of medical imaging [66], but also starts to trickle into the world of parameterized PDE models: Raissi et al. presented in [100] a new method to utilize neural networks to solve PDE equations by means of introducing the PDE equations as a "soft constraint". This kind of PDE-penalizing optimization is not new and has already been done in other works (e.g. [121]), but the novelty of the approach comes from the parameterized model: PINNs use a neural network as their underlying model from which they can apply back-propagation techniques, commonly found in machine learning frameworks, to directly fit the neural network to the PDE. The presenting problem is highly non-linear and non-convex, depending on the chosen network architecture and is therefore solved using the current state-of-the-art solver for machine learning problems, namely Adaptive Moment Estimation (ADAM) [79]. In Section 4.6.2, we compare our model presented in [61] to optimize conductivities in atrial-endocardial maps, to a PINN approach called EikonalNet [109]. Several beneficial modeling assumptions were made in [61] that helped the optimization in better finding the conduction velocity tensors. These assumptions were carried over to EikonalNet in our paper [60], where we use a PINN to model and optimize the same problem. The comparison to PIEMAP showed that in the synthetic/in-silico test-case, PIEMAP performed better, but the PINN approach has also some advantages: The main benefits are, first that we do not need to specifically choose the initiation sites of the eikonal solution, and secondly that the model does not strictly enforce the eikonal equation, which can be a mixed blessing (see Section 4.6.3 for more details).

In this work, we will use the (comparably) computationally efficient anisotropic eikonal method as a baseline physical model for electrical propagation. The equation itself was already extensively discussed in Section 2.8.1. It has been shown to be a good compromise between physiological accuracy and computational cost [33] and thus meets our requirements for modeling the inverse problem. Multiple methods have been developed to efficiently solve the eikonal equation: In the case of the isotropic eikonal equation, e.g. $D = c\mathbf{I}$, the FMM has a very favorable computational complexity of $\mathcal{O}(n \log n)$ [112, 118]. The original FIM on grids presented in [73] has a worse computational complexity, but has been shown to converge faster in many practical examples as it can heavily exploit parallelism, commonly found on modern machines, especially GPUs. At the date of writing, arguably one of the most popular methods are the extensions of the original FIM [52, 53] that extended the algorithm to the anisotropic eikonal equation and mesh domains. In [52], the method was first extended to general triangulated surfaces which we will later employ in our algorithm PIEMAP in Section 4.6.2. The extension to tetrahedral meshes in [53] provided a good framework for many methods presented in this thesis in this chapter. The exact update steps of the latter method will be discussed in detail in Section 4.3 and will later be used in Section 4.5 for the gradient computation.

Recall the anisotropic eikonal equation of Section 2.8.1:

$$\begin{cases} \|\nabla u\|_D = \sqrt{\langle D\nabla u, \nabla u\rangle} = 1 & \text{in } U \\ u = g & \text{on } \Gamma \subset \partial U \end{cases} \qquad \text{((2.23) revisited)}$$

where we require $D \in S^n_{++}$

$$S^n_{++} = \left\{ A | A = A^\top \wedge \mathbf{x}^\top A \mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^n \setminus \mathbf{0} \right\}. \qquad \text{((2.6) revisited)}$$

Upon inspecting (2.23) we can immediately identify the two main parameters of the inverse problem: The conductivity tensor D and the EASs, defined by their location, timing, as well as their number (i.e. the Dirichlet boundary conditions). The problem of fitting these parameters to clinical data has already been considered in the literature for the conduction velocities in our works [59, 61] and others [8, 114]. However, the optimization of EASs received limited attention so far with only a few works dealing with activation onsets [75] or locations [80]. The simultaneous optimization of EASs (especially their number) and conduction velocity has been analyzed only very recently in [56, 59, 96].

4.2 Methodological Foundations

As already discussed in Section 3.1, the eikonal equation has strong links to cardiac activation under minor simplifying assumptions. The tensor D of (2.23) directly encodes the squared conduction velocity tensor. As conduction velocities in the ventricular myocardium are known to be orthotropic with typical velocity ratios around $v_f : v_s : v_n \approx 0.6 : 0.4 : 0.2\, m/s$ [33], the anisotropic version of the eikonal equation will be used in all scenarios. Many formulations and other works [9, 115] consider the physiological concepts of fibers, sheet and normal direction as well as their respective velocities. Since we require $D \in S^n_{++}$ it turns out that the conductivity tensor encodes these properties in its eigenvalues and eigenvectors. In the presented formulation, D can be defined through the fiber, sheet and normal direction as:

$$D(\mathbf{x}) \coloneqq v_f^2(\mathbf{x})\, \mathbf{f}(\mathbf{x}) \otimes \mathbf{f}(\mathbf{x}) + v_s^2(\mathbf{x})\, \mathbf{s}(\mathbf{x}) \otimes \mathbf{s}(\mathbf{x}) + v_n^2(\mathbf{x})\, \mathbf{n}(\mathbf{x}) \otimes \mathbf{n}(\mathbf{x}), \tag{4.1}$$

Here, D encodes the squared conduction velocities along the tissue's eigenaxes where $\mathbf{f}(\mathbf{x}), \mathbf{s}(\mathbf{x})$ and $\mathbf{n}(\mathbf{x})$ are the fiber, sheet and normal vectors respectively and $v_f(\mathbf{x}), v_s(\mathbf{x})$ and $v_n(\mathbf{x})$ encode their respective propagation velocities. Equation (4.1) is in reality a simple reformulation of the eigendecomposition of D in (2.5)

$$D = \underbrace{\begin{bmatrix} f_1 & s_1 & n_1 \\ f_2 & s_2 & n_2 \\ f_3 & s_3 & n_3 \end{bmatrix}}_{U} \underbrace{\operatorname{diag}\left(v_f^2, v_s^2, v_n^2\right)}_{\Sigma} \underbrace{\begin{bmatrix} f_1 & f_2 & f_3 \\ s_1 & s_2 & s_3 \\ n_1 & n_2 & n_3 \end{bmatrix}}_{U^\top} \tag{4.2}$$

In order to solve (2.23), many commonly encountered methods, such as the FIM in [53] rely on solving the Hamilton–Jacobi formulation, briefly touched in Section 2.8.1.

$$\begin{cases} \dot{\mathbf{x}} = \nabla_{\mathbf{p}} H(\mathbf{p}, \mathbf{x}) \\ \dot{\mathbf{p}} = -\nabla_{\mathbf{x}} H(\mathbf{p}, \mathbf{x}). \end{cases} \qquad \text{((2.27) revisited)}$$

Before we start with the proposed methods, we will shortly summarize the FIM.

4.3 Solving the Anisotropic Eikonal Equation - The Fast Iterative Method (FIM)

The FIM is an efficient method to solve the anisotropic eikonal equation and variants have been presented for grids [73], triangular [52] and tetrahedral [53] domains. We will consider here the tetrahedral variant. At its heart is the local update rule, computing a vertex value by computing the travel time from the opposite face of the three other vertex values inside a tetrahedron. For this purpose, the 3D-domain Ω is approximated using Lagrangian $\mathcal{P}_1$ elements. For practial purposes, we define $M = \{V, T\}$ to be the mesh, consisting of a discrete set of vertices $V = \bigcup_i \mathbf{v}_i \in \mathbb{R}^3$, combined with a discrete set of tetrahedra $T = \bigcup_j T_j$, consisting of four vertices each $T_j = (\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3}, \mathbf{v}_{j_4})$. Note that in the algorithm, D is assumed to be piecewise constant ($\mathcal{P}_0$), i.e. there is exactly one constant diffusion Tensor D_j associated with each T_j. For convenience the minimum arrival time associated to a vertex $\mathbf{v}_i$ is written as $\phi(\mathbf{v}_i) = \phi_i$.

An especially useful practical feature of Lagrangian $\mathcal{P}_1$ elements is that each point in the simplexes spanning the domain can be uniquely described by its barycentric coordinates Δ_k, which apparently are equal to its basis functional evaluation (see Section 2.9 for more details).

$$\Delta_k = \left\{ (\lambda_1, \dots, \lambda_k)^\top \middle| \sum_{i=1}^{k} \lambda_i = 1 \wedge \lambda_i \geq 0 \right\} \qquad \text{((2.46) revisited)}$$

Note that simlarly any point on the triangle spanned by $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ opposite of a vertex can be described using three-dimensional barycentric coordinates.

The local update rule inside one tetrahedron, which defines the travel time from any point on the face $\mathbf{x}_{1,2,3}$ to the vertex $\mathbf{v}_4$, is then given by

$$\phi_4 - \phi_5 = \left|\left|\mathbf{v}_4 - \sum_{i=1}^{3} \lambda_i \mathbf{v}_i\right|\right|_{2,D_j^{-1}} = ||\mathbf{e}_\Delta||_{2,D_j^{-1}} = \sqrt{\mathbf{e}_\Delta^\top D_j^{-1} \mathbf{e}_\Delta} > 0,$$

which is associated length with the linear curve through the element. In order to update a single node $\mathbf{v}_4$, given all other vertex values inside a single tetrahedron ϕ_1, ϕ_2, ϕ_3, we want to find the point $\mathbf{x}_{1,2,3}$ that minimizes the travel time from the face, to the vertex $\mathbf{v}_4$ in the metric D_j. This can be formulated as the minimization problem:

$$\phi_4^I(\phi_1, \phi_2, \phi_3) = \min_\lambda \sum_{i=1}^{3} \lambda_i \phi_i + \sqrt{\mathbf{e}_\Delta^\top D_j^{-1} \mathbf{e}_\Delta}, \text{ s.t. } \lambda \in \Delta_3 \tag{4.3}$$

The computation is visualized in Figure 4.1. If we set $\lambda_3 = 1 - \lambda_1 - \lambda_2$ and solve

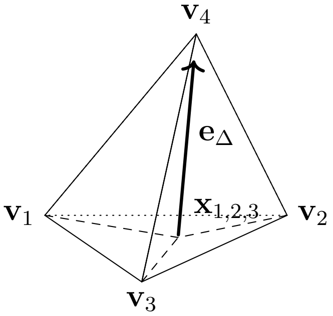


Figure 4.1: Local update of the FIM solver within one tetrahedron. The update is computed using the linear distance formulation $\mathbf{e}_\Delta$ from the point $\mathbf{x}_{1,2,3}$ on the face between nodes $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ to the point $\mathbf{v}_4$ to be computed.

Equation (4.3) w.r.t. one λ_i, while the other barycentric coordinate is called λ_k, then the optimal solution to the unconstrained problem is the solution to the quadratic problem

$$\lambda_{i_{1,2}} = \frac{-p_{1,2} \pm c\sqrt{\frac{p_{1,1}p_{2,2} - p_{1,2}^2}{p_{1,1} - c^2}}}{p_{1,1}} \tag{4.4}$$

with

$$\begin{aligned}
\mathbf{w}_1 &= \mathbf{v}_3 - \mathbf{v}_1; \quad \mathbf{w}_2 = \mathbf{v}_3 - \mathbf{v}_2; \quad \mathbf{w}_3 = \mathbf{v}_4 - \mathbf{v}_3 \\
\mathbf{z}_1 &= \mathbf{w}_i - \frac{\mathbf{w}_2^\top D^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)}{\mathbf{w}_j^\top D^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)} \mathbf{w}_j \\
\mathbf{z}_2 &= \mathbf{w}_3 - \frac{\mathbf{w}_3^\top D^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)}{\mathbf{w}_j^\top D^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)} \mathbf{w}_j \\
c &= \phi_i - \phi_3 - \frac{\mathbf{w}_i^\top D^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)}{\mathbf{w}_j^\top D^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)} \\
p_{a,b} &= \mathbf{z}_a^\top D^{-1} \mathbf{z}_b
\end{aligned} \tag{4.5}$$

Once we found λ_i, computing λ_k for $k \neq i$ becomes:

$$\lambda_k = -\frac{\left(\mathbf{w}_3 + \lambda_i \mathbf{w}_i\right)^\top D_j^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)}{\mathbf{w}_j^\top D_j^{-1} \left(\mathbf{w}_2(\phi_1 - \phi_3) - (\phi_2 - \phi_3)\mathbf{w}_1\right)} \tag{4.6}$$

The derivation that arrives at this conclusion is given in Section A.2. The actual minimal solution of λ_1 and λ_2 in Equation (4.3) is a constrained quadratic minimization problem, for which special cases may arise when solving it unconstrained according to Equation (4.4) as Fu et al. already stated:

> *If no root exists, or if λ_1 or λ_2 falls outside the range of $[0, 1]$ (that is, the characteristic direction does not reside within the tetrahedron), we then apply the 2D local solver used in [52] to the faces $\Delta_{1,2,4}$, $\Delta_{1,3,4}$ and $\Delta_{2,3,4}$ and select the minimal solution from among the three.* ([53])

where the mentioned root refers to the root term in Equation (4.4). The referenced solution for the 2D-case behaves similarly (see [59] for more details). $\mathbf{v}_4$ is in most cases part of multiple tetrahedra and the minimum arrival time ϕ_4 is therefore the minimum arrival time over all tetrahedra T_j that $\mathbf{v}_4$ is a vertex of ($\mathbf{v}_4 \in T_j$):

$$\phi_4 = \min_j \phi_4^*(\phi_1, \phi_2, \phi_3) \text{ s.t. } \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4 \in T_j \tag{4.7}$$

To globally compute the eikonal solution, FIM initializes all vertices with starting times to their value and adds their neighbors to the Active List (AL), while all other arrival times are set to ∞. The AL keeps track of all points to be updated according to Equation (4.7) in the next iteration. Converged points are removed from the active list, while points are added that received an update. If one or more of ϕ_1, ϕ_2 and ϕ_3 are not yet computed (meaning their value is still ∞ from the initialization), (4.3) and (4.7) still work if we define $0 \cdot \infty = 0$ for this sake. Once the value ϕ_4 has been updated, all points sharing a tetrahedron with v_4 are recomputed using (4.7) and are added to the AL if their activation time decreased. The convergence property of the FIM algorithm removes a point from the AL if the change between two subsequent iterations, k and $k+1$, is smaller than ε: $\left|\phi_4^k(\mathbf{x}) - \phi_4^{k+1}(\mathbf{x})\right| < \varepsilon$. FIM can be efficiently computed on Single Instruction Multiple Data (SIMD) architectures, such as GPUs, thanks to the easy parallelization of the updates of all points in the AL. Relaxing constraints on ε can be used to reduce computational costs at the expense of precision, or to avoid small rounding errors due to limited floating point precision. An outline of the computations done by the FIM are given in Algorithm 4.1. Note that this special variant of the algorithm also keeps track of all intermediate updates, later used in Section 4.5.

Algorithm 4.1: Fast Iterative Method (FIM) [53]

```
Input  : Diffusion Tensors $D$, Starting Points $S$, Mesh $M$ with Neighborhood
         Operator $\mathcal{N}$ Initiation Timings Function $g(\mathbf{v})$
Output: First arrival times $\phi$, FIM Updates $\text{FIM}_U$
$\forall \mathbf{v}_0 \in V(S) : \phi(\mathbf{v}_0) := g(\mathbf{v}_0)$
$\forall \mathbf{v} \in V(M) \setminus S : \phi(\mathbf{v}) := \infty$
$\text{AL} = \bigcup_{\forall \mathbf{v}_0 \in V(S)} \mathcal{N}(\mathbf{v}_0)$
$\text{FIM}_U = \varnothing$
while $AL \neq \varnothing$ do
    for $\mathbf{v}_i \in AL$ do
        $\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l \in \mathcal{N}(\mathbf{v}_i)$
        $\phi_{\text{new},i} := \phi_4(\phi_j, \phi_k, \phi_l)$ // Compute local eikonal solution (Equation (4.7))
        $\phi_{\text{old},i} := \phi(\mathbf{v}_i)$
        $\phi(\mathbf{v}_i) := \phi_{\text{new},i}$
        $\text{FIM}_U = \text{FIM}_U \cup (\phi_{\text{new},i}, \mathbf{v}_4, \lambda)$
        // If point has converged
        if $|\phi_{new,i} - \phi_{old}| < \varepsilon$ then
            $\text{AL} = \text{AL} \setminus \mathbf{v}_4$ // Remove converged from AL
            // Check if neighboring points are converged
            for $\forall \mathbf{v}_n \in \mathcal{N}(\mathbf{v}_i)$ do
                $\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l \in \mathcal{N}(\mathbf{v}_i)$
                $\phi_{\text{new},n} := \phi_4(\phi_j, \phi_k, \phi_l)$ // Equation (4.7)
                if $|\phi_{new,n} - \phi(\mathbf{v}_n)| \geq \varepsilon \wedge \phi_{new} < \phi(\mathbf{v}_n)$ then
                    $\text{AL} = \text{AL} \cup \mathbf{v}_n$ // Add to AL if not converged
                end
            end
        end
    end
end
```

Remark 4.1. *Note that the formulation of computing the travel time, i.e. the distances in the metric, internally relies again on the Hamilton–Jacobi formalism (see Sections 2.8 and 4.7). Thus, the update rule uses the inverse metric D^{-1}.*

4.4 Runtime

An important consideration for clinical applicability when talking about algorithms, is their runtime. In practice, it would desirable to have inverse methods that can estimate parameters within less than a few hours to meet clinical demands, but this is in practice very difficult to achieve as we need to solve the underlying PDEs and ODEs several times. However, one factor that we can exploit is the rise of modern hardware architectures of the past decade, particularly GPUs. GPGPU computing has equipped individual researchers, with computing capabilities, formerly only known to whole research groups or institutions in the form of High Performance Computing (HPC) clusters. Figure 4.2 shows how the theoretical Floating Point Operations per Second (FLOPs/s) of single user GPUs have grown. Though GPUs only efficiently work on a small subset of problems that can efficiently exploit their SIMD architecture (a drawback that does not apply to many HPC architectures), especially many modern ML frameworks are fueled and are now available to a broad user base thanks to the advent of GPU computing. Most of the proposed methods in this chapter work on the said GPU architectures and thus benefit from the constant development and evolution of modern GPU hardware. This allowed us to run all of the proposed methods on a single desktop machine, rather than an HPC cluster, though incorporating multiple machines could yield potential speedups. All of the results in this chapter were therefore computed on a desktop machine with an Intel Core i7-5820K CPU with 6 cores of each 3.30GHz, 32GB of working memory and a NVidia RTX 2080 GPU. The runtimes of all methods and experiments differ greatly and are listed individually in the result section of each.

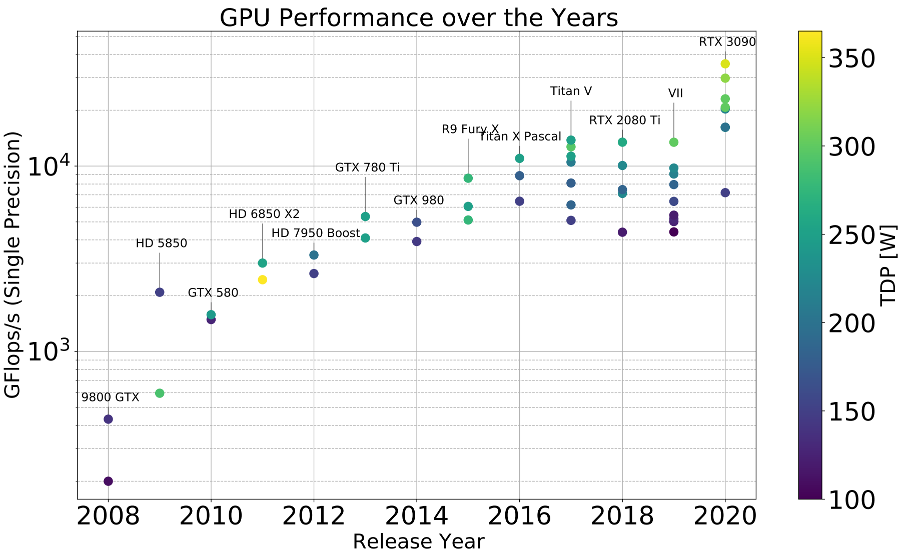


Figure 4.2: Growth of the theortically achievable FLOPs/s for various consumer GPUs over the course of the years.

4.5 FIMIN: A direct approach at solver-based gradient computation

In [59], we introduced a gradient-based optimization, based on the FIM in [53], further detailed in Section 4.3.

Computing $\phi(\mathbf{x})$ given $g(\mathbf{x})$ and $D(\mathbf{x})$ is called the forward anisotropic eikonal problem. In contrast, computing $g(\mathbf{x})$ and/or $D(\mathbf{x})$ from data $\phi(\mathbf{x})$ observed on a subdomain Ω_O is called the inverse anisotropic eikonal problem. Except for very restricted cases, the general inverse eikonal problem is considered ill-posed.

To find D and g, we start with a common optimization approach

$$\begin{aligned}
&\min_{D,g} \int_{\Omega_\text{O}} \left(\phi(\mathbf{x}) - \hat{\phi}(\mathbf{x})\right)^2 \mathrm{d}\mathbf{x} \\
&\text{s.t. } D(\mathbf{x}) \in S^n_{++}, ||\nabla\phi(\mathbf{x})||_{2,D^{-1}(\mathbf{x})} = 1 \quad && \forall \mathbf{x} \in \Omega \\
&\phantom{\text{s.t. }} \phi(\mathbf{x}_0) = g(\mathbf{x}_0) && \forall \mathbf{x}_0 \in S
\end{aligned} \tag{4.8}$$

where Ω_O refers to the observation domain and $\hat{\phi}(\mathbf{x})$ are the observed arrival times in Ω_O. Equation (4.8) states that we want to minimize the squared distance of computed arrival times $\phi(\mathbf{x}) - \hat{\phi}(\mathbf{x})$ over the observable domain Ω_O, assuming that activation times $\phi(\mathbf{x})$ are compatible with the anisotropic eikonal condition in (2.23) and that $D(\mathbf{x})$ is in the set of symmetric positive definite matrices on the whole domain Ω

4.5.1 Inverse Problem - Gradient Computation

For the inverse solution of the eikonal equation, where the unknown values for $g(\mathbf{x})$ and S are solved from a known solution $\phi(\mathbf{x})$, the gradient of FIM is required. Additionally, the local update rules Equation (4.3) and Equation (4.4) for each point must be further considered. In order to derive the gradient w.r.t. an optimizable variable ϑ, we define all nodal values $\phi_i(\vartheta)$ and diffusion tensors $D_j(\vartheta)$ as functions of ϑ. Given the optimal choice of λ_1 and λ_2, the derivative of (4.3) w.r.t. ϑ and a diffusion tensor function $D(\mathbf{x}, \vartheta) = D_j$ of the tetrahedron $\mathbf{x} \in T_j$ becomes:

$$\frac{\partial \phi_4}{\partial \vartheta}(\phi_1, \phi_2, \phi_3) = \begin{cases} \sum_{i=1}^3 \left(\frac{\partial \lambda_i}{\partial \vartheta}\phi_i + \lambda_i \frac{\partial \phi_i}{\partial \vartheta}\right) + \frac{1}{2\sqrt{\mathbf{e}_\Delta^\top D_j \mathbf{e}_\Delta}}\left(2\frac{\partial \mathbf{e}_\Delta^\top}{\partial \vartheta} D_j \mathbf{e}_\Delta + \mathbf{e}_\Delta^\top \frac{\partial D_j}{\partial \vartheta}\mathbf{e}_\Delta\right) & \text{if } \lambda \in \Delta_3 \\ \sum_{i=1}^2 \left(\frac{\partial \lambda_i}{\partial \vartheta}\phi_i + \lambda_i \frac{\partial \phi_i}{\partial \vartheta}\right) + \frac{1}{2\sqrt{\mathbf{e}_\Delta^\top D_j \mathbf{e}_\Delta}}\left(2\frac{\partial \mathbf{e}_\Delta^\top}{\partial \vartheta} D_j \mathbf{e}_\Delta + \mathbf{e}_\Delta^\top \frac{\partial D_j}{\partial \vartheta}\mathbf{e}_\Delta\right) & \text{if } \lambda \in \Delta_2 \\ \frac{\partial \lambda_i}{\partial \vartheta}\phi_i + \frac{1}{2\sqrt{\mathbf{e}_\Delta^\top D_j \mathbf{e}_\Delta}}\left(2\frac{\partial \mathbf{e}_\Delta^\top}{\partial \vartheta} D_j \mathbf{e}_\Delta + \mathbf{e}_\Delta^\top \frac{\partial D_j}{\partial \vartheta}\mathbf{e}_\Delta\right) & \text{else} \end{cases} \tag{4.9}$$

with $\frac{\partial \mathbf{e}_\Delta}{\partial \vartheta} = \mathbf{v}_4 - \sum_{i=1}^3 \frac{\partial \lambda_i}{\partial \vartheta}\mathbf{v}_i$. In order to derive Equation (4.4), we may need to derive each variable of the quadratic problem, assuming dependency of each variable on ϑ. The exact derivations of each of the variables can be found in Section A.3.

The derivation $(\partial\phi_i)/(\partial\vartheta)$ is recursively acquired by applying the chain rule (i.e. backpropagation) along the path of updates we computed in the forward FIM solution, until we computed all necessary updates. For the case that the solution of Equation (4.7) of several tetrahedra share the same arrival time, there is no single unique solution for $\frac{\partial \lambda_i}{\partial \vartheta}$. Recall for this purpose the superdifferential

$$\nabla^+\phi(\mathbf{x}) := \left\{\mathbf{p} \in \mathbb{R}^n : \limsup_{\mathbf{y}\to\mathbf{x}} \frac{\phi(\mathbf{y}) - \phi(\mathbf{x}) - \langle \mathbf{p}, \mathbf{y}-\mathbf{x}\rangle}{||\mathbf{y}-\mathbf{x}||} \leq 0\right\} \qquad \text{((2.11) revisited)}$$

Our optimization uses the superdifferential $\nabla^+\phi$ in these cases, which can be interpreted as the supporting hyperplane above the minimum functions in (4.7). These superdifferentials become relevant inside elements where multiple wavefronts arrive at the same time (Compare Example 2.4). Since any choice of superdifferential $\mathbf{p} \in \nabla^+\phi$ is feasible for the optimization (see Section 2.10), we choose a random superdifferential.

The FIM algorithm is a Gauss-Seidel method, meaning that nodes will be updated multiple times until the final, minimum arrival time is calculated. Up until now, we only considered this final value, which is sufficient for the forward problem. However, for the differentiation of ϕ_4 in Equation (4.9), non-trivial cases may occur that might require the results of previous updates of ϕ_4. Consider a simple example 2D-mesh with 3 triangles and 4 nodes v_1 through v_4 and their respective arrival times ϕ_1 through ϕ_4, visualized in Figure 4.3. On the left side, the update directions (similar to $\mathbf{e}_\Delta$ in Equation (4.3), but in 2D) are marked as arrows for the first update of each node. On the right side of Figure 4.1, we now assume that ϕ_2 receives an update from ϕ_4, which happens if the characteristic directions do not coincide with the edges and faces of the mesh. This update of ϕ_2 will result in possibly additional updates of ϕ_3, ϕ_4 and subsequently ϕ_2, until convergence according to $\left|\phi_4^k(\mathbf{x}) - \phi_4^{k+1}(\mathbf{x})\right| < \varepsilon$ is reached. If we assume $D \in S^2_{++}$ for all triangles, then the update directions can never form a full circle, since for any update direction it holds that $\forall \mathbf{e}_{i,j} \neq \mathbf{0} : ||\mathbf{e}_{i,j}||_{2,D} > 0$. To derive $\frac{\partial\phi_2}{\partial\vartheta}$ properly, we need to distinguish between different updates of ϕ_2. Let ϕ_i^k denote the k-th update of ϕ_i and $f(\phi_i, \phi_j) =$ Equation (4.3) (FIM Update). Then (assuming $\phi_0^1 = 0$):

$$\phi_2^2 = f(\phi_3^1, \phi_4^1) = f(f(\phi_1^1, \phi_2^1), f(\phi_2^1, \phi_3^1)) = f(f(\phi_1^1, f(0, \phi_1^1)), f(f(0, \phi_1^1), \phi_3^1)) \tag{4.10}$$

To this end, we use a forward method of differentiation, beginning from the starting points $\mathbf{x}_0 \in S$ and applying the gradient in the same update order as the original computation of ϕ_j to compute $\frac{\partial\phi_j}{\partial\vartheta}$. This way, all ϕ_j-values reflect the exact values and differentials at any given iteration of the algorithm. Instead of computing the forward solution a second time, we remember 250 Gauss-Seidel iterations for this purpose by using ring-buffers, which was sufficient for all experiments conducted. To avoid duplicate evaluation, we remember the values of λ_i from (4.3) for each point and the vertices $\mathbf{v}_j$ as well as their values ϕ_j at the time of the minimal evaluation. The backpropagation starts at the starting points $\mathbf{v}_0$ and applies the chain rule along the path of updates we computed in the forward

FIM solution. Remembering the Gauss-Seidel updates increases the required memory, but in return speeds up the computation. This however may lead to very high memory consumption for large meshes, in which case an implementation that computes the forward solution and all derivatives jointly would be preferrable.

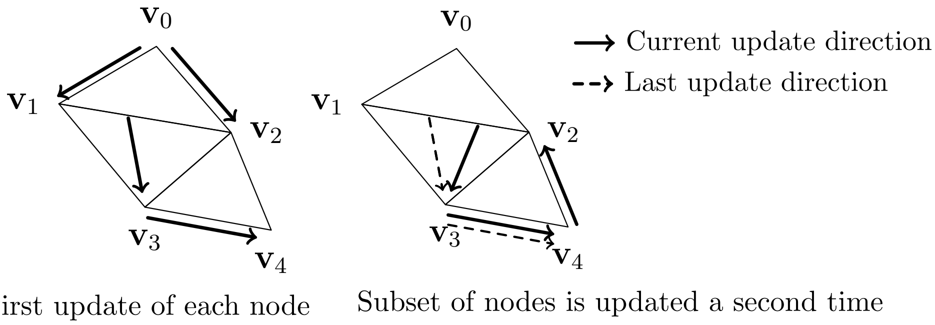


Figure 4.3: Visualization of an exemplary update sequence of the FIM algorithm that demonstrates the necessity to distinguish between nodal values of different iterations. The arrows visualize the update directions similar to $\mathbf{e}_\Delta$ in Equation (4.3). the High anisotropies may lead to multiple updates of nodes with an update direction that is dependant on itself. For such cases, we need to remember the earlier computations of the same node. Note that the update directions can never form a full circle if all diffusion tensors are symmetric and positive definite.

4.5.2 Optimization

Moving from the continuous optimization problem in (4.8) to the discrete domain, we define our main optimization problem using an arbitrary given function $\hat{\phi}$ on a measurable discrete surface Ω_O:

$$\begin{aligned} &\min_{D,\mathbf{g}} \frac{1}{2} \sum_{\mathbf{v}\in\Omega_\mathrm{O}} \left(\phi(\mathbf{v}) - \hat{\phi}(\mathbf{v})\right)^2 + \gamma\mathcal{R}\left(D, \mathbf{g}\right) \\ &\text{s.t. } D(\mathbf{x}) \in S^n_{++}, \phi(\mathbf{x}) = \mathrm{FIM}(D, \mathbf{g}, M, S)(\mathbf{x}) \end{aligned} \tag{4.11}$$

where $\phi(\mathbf{y})$ and $\hat{\phi}$ are the vector of nodal vertex values ϕ_i from the current model and reference respectively and $\mathcal{R}$ is a regularizing function, penalizing unlikely solutions. We refer to the sum as the Sum of Squared Errors (SSE).

To properly derive all final points $\frac{\partial \phi_i}{\partial \vartheta}$, we apply all saved updates FIM_U in their original order and derive each updated value using Equation (4.9). The computation takes only a fraction of the original time of the FIM algorithm, since a lot of computations in the original FIM algorithm are dedicated to checking if the values of vertices have converged. An outline of the procedure is given in Algorithm 4.2.

All that is left for our optimization in Equation (4.11) to work, is to define a proper regularization function $\mathcal{R}$. The proper choice of $\mathcal{R}$ has been a long debated topic in multiple fields for inverse problems, but we used an arguably very popular choice for regularzing the diffusion tensors D, that has been successfully used in many inverse problems, such as image denoising: The Huber regularization [107], a smoothed Total-Variation function, $(\mathcal{R}(D, \mathbf{g}) = ||\nabla D||_{H,\alpha})$, where $||\cdot||_{H,\alpha}$ denotes the Huber norm defined as:

$$||\mathbf{x}||_{H,\alpha} = \begin{cases} \frac{1}{2}\,||\mathbf{x}||^2 & \text{if } ||\mathbf{x}|| \leq \alpha \\ \alpha\left(||\mathbf{x}|| - \frac{1}{2}\alpha\right) & \text{else} \end{cases}$$

To compute ∇D we use Gauss' theorem since D is defined as a piecewise constant function. Details how we approximated this gradient are given in Section A.4.

Several algorithms are suitable to optimize our objective function in (4.11), but the number of iteration steps within a given amount of time is severely limited by the size of the mesh and the number of parameters. The original problem in Equation (4.11) can be reformulated as the non-linear least squares problem, given the parameter set $\mathbf{y} \in \mathbf{C}$, where $\mathbf{C}$ is a convex constraint set:

$$\min_{\mathbf{y}\in\mathbf{C}} \frac{1}{2}\left|\left|\phi(\mathbf{y}) - \hat{\phi}\right|\right|^2_{2,\Omega_\mathrm{O}} + \frac{\gamma}{2}\sum_{j=1}^{|T|}\left|\left|(\nabla\mathcal{D}\mathbf{y})_j\right|\right|_{H,\alpha} = \min_{\mathbf{y}\in\mathbf{C}} \frac{1}{2}\,||\mathbf{r}(\mathbf{y})||^2_{2,\Omega_\mathrm{O}} + \frac{\gamma}{2}\sum_{j=1}^{|T|}\left|\left|(\nabla\mathcal{D}\mathbf{y})_j\right|\right|_{H,\alpha} \tag{4.12}$$

with γ being the regularization weight. Only errors on the measurement domain Ω_O contribute to the norm $||\cdot||^2_{2,\Omega_\mathrm{O}} = \sum_{\mathbf{v}\in\Omega_\mathrm{O}} (\cdot)^2$. $\mathcal{D}$ is our linear diffusion tensor assembly operator, creating the tensors D from the elements in the parameter vector $\mathbf{y}$. We define the gradients on the diffusion tensors as the sum of variation of the tensor's elements:

$$(\mathcal{D}(\mathbf{y}))_j = \begin{bmatrix} a & d & e \\ d & b & f \\ e & f & c \end{bmatrix} \qquad \left|\left|(\nabla\mathcal{D}(\mathbf{y}))_j\right|\right| = \sum_{\xi\in\{a,\ldots,f\}} ||\nabla\xi||$$

where j denotes the diffusion tensor D_j of T_j and each element is a function of $\mathbf{y}$.

We use a constrained Gauss-Newton method to optimize Equation (4.12), by linearizing the function around the current value $\mathbf{y}_k$ using a first order Taylor series expansion and discarding higher order terms:

$$\min_{\mathbf{y}\in\mathbf{C}} \frac{1}{2}\,||\mathbf{r}(\mathbf{y})||^2 + \frac{\gamma}{2}\sum_{j=1}^{|T|}\left|\left|(\nabla\mathcal{D}(\mathbf{y}))_j\right|\right|_{H,\alpha} \approx \min_{\mathbf{y}\in\mathbf{C}} \frac{1}{2}\,||\mathbf{r}(\mathbf{y_k}) + \mathbf{J}(\mathbf{y}_k)\,(\mathbf{y} - \mathbf{y}_k)||^2 + \frac{\gamma}{2}\sum_{j=1}^{|T|}\left|\left|(\nabla\mathcal{D}(\mathbf{y}))_j\right|\right|_{H,\alpha} \tag{4.13}$$

where $\mathbf{J}(\mathbf{y}_k)$ denotes the Jacobian matrix of $\mathbf{r}$ at the point $\mathbf{y}_k$.

The size of the Jacobian easily exceeds the available memory, but since only few of the activation times are dependant on single variables and diffusion tensors, the Jacobian matrix can be represented in a memory-efficient sparse matrix format.

Algorithm 4.2: Computing FIM Gradients

```
Input  : Diffusion Tensor Operator $\mathcal{D}(\mathbf{y})$, Starting Points $S$, Mesh $M$
         Activation Times $\phi$
         Differential Operator $\partial\mathcal{D}$
         FIM Updates (Equation (4.7)) $\mathrm{FIM}_U$
Output: Activation Times Derivation $\frac{\partial\phi}{\partial\mathcal{D}}$
$\forall \mathbf{v}_j \in M \setminus S : \phi_j := \infty$
$\forall \mathbf{v}_j \in M : \frac{\partial\phi_j}{\partial y_i} := 0$
for $\phi_u, \mathbf{v}_n, \lambda \in FIM_U$ do
    $\phi_n := \phi_u$    // Reapply Update
    $\frac{\partial\phi_n}{\partial\mathcal{D}} := f(\frac{\partial\mathcal{D}}{\partial y_i}, \mathcal{D}(\mathbf{y}), \phi, \frac{\partial\phi}{\partial y_i}, \lambda)$    // Calculate $f$ = Equation (4.9)
end
```

Algorithm 4.3: FIMIN

```
Input  : Initial Parameter set $\mathbf{y}_0 \in \mathbb{R}^K$
         Diffusion Tensor Operator $\mathcal{D}(\mathbf{y})$
         Convex Constraint Set $\mathbf{C}$
         Differential Operator $\frac{\partial\mathcal{D}}{\partial y}$
         Desired Activation Map $\hat{\phi}$
Output: Optimized Parameter Set $\mathbf{y}_n$
while not converged do
    $\mathbf{r}(\mathbf{y_k}) = \phi\left(\mathcal{D}(\mathbf{y}_k)\right) - \hat{\phi}$    // Compute Residuals
    $i \in V(M) : \mathbf{J}_{i,j}(\mathbf{y}_k) = \frac{\partial\phi_i}{\partial y_j}$    // Build Jacobian using Algorithm 4.2
    $\hat{\mathbf{y}}_{k+1} = \min_{\mathbf{y}\in\mathbf{C}} \frac{1}{2} \left|\left|\mathbf{r}(\mathbf{y_k}) + \mathbf{J}(\mathbf{y}_k)\left(\mathbf{y} - \mathbf{y}_k\right)\right|\right|^2_{2,\Omega_O} + \frac{\gamma}{2} \sum_{j=1}^{|T|} \left|\left|\left(\nabla\mathcal{D}\mathbf{y}\right)_j\right|\right|_{H,\alpha}$
        // Constr. Gauss-Newton
    $\mathbf{y}_{k+1} = \mathbf{y}_k - \beta\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k+1}\right)$
end
```

Algorithm 4.2 is also applicable to optimize the activation timings $\mathbf{g}$ if the operator $\mathcal{D}(\mathbf{y})$ is swapped with $\mathbf{g}(\mathbf{y})$. The constrained Gauss-Newton optimization problem in Equation (4.13) is a convex problem, which we solve by using a bounded L-BFGS method [21]. In Algorithm 4.3, the activation times derivation and the constrained Gauss-Newton method are combined to iteratively adapt the parameter set to the minimize Equation (4.11). The step-size parameter β is found by using the Armijo backtracking line search [4].

The L-BFGS method is limited to box constraint sets $\mathbf{C_b}$, but, as shown in Example 2.10, the problem can also be solved for any convex constraint $\mathbf{C}$ with an unconstrained optimization algorithm by defining the Moreau-enveloped problem of Equation (4.13). We define in this case:

$$f(\bar{\mathbf{y}}) := \min_{\mathbf{y}} \frac{1}{2} ||K\mathbf{y} - \mathbf{b}||^2_{2,\Omega_\mathrm{O}} + \delta_{\mathbf{C}}(\mathbf{y}) + \frac{1}{2} ||\mathbf{y} - \bar{\mathbf{y}}||^2_{2,M} + \frac{\gamma}{2} ||A\mathbf{y}||^2_2 \tag{4.14}$$

where $\delta_{\mathbf{C}}(\mathbf{x})$ is the indicator function on $\mathbf{C}$ and $M = \frac{1}{\tau}I - K^\top K - \gamma A^\top A$. The Huber regularization can not be used for the regularization term $\frac{\gamma}{2} ||A\mathbf{y}||^2_2$ without any modifications. For this purpose, we approximate the Huber H function locally around the current point $\mathbf{y}_k$ by defining

$$||\mathbf{x}||_{H_k,\alpha} = \begin{cases} \frac{1}{2} ||\mathbf{x}||^2 & \text{if } ||\mathbf{x}_k|| \leq \alpha \\ \alpha \left(\frac{||\mathbf{x}||^2}{||\mathbf{x}_k||} - \frac{1}{2}\alpha \right) & \text{else} \end{cases}$$

Our total regularization operator is therefore $A = H_k \nabla \mathcal{D}$. $M \in S^n_{++}$ is ensured by $\tau < \left(||K^\top K|| + \gamma ||A^\top A|| \right)^{-1}$ (see (2.10)). The optimal choice of $\mathbf{y}$ in (4.14), yields a proximal point algorithm returning a feasible point:

$$\hat{\mathbf{y}} = \operatorname*{proj}_{\tau\mathbf{C}} \left(\bar{\mathbf{y}} - \tau K^\top (K\bar{\mathbf{y}} - \mathbf{b})_{\Omega_\mathrm{O}} - \tau\gamma A^\top A\bar{\mathbf{y}} \right)$$

where $\mathrm{proj}_{\tau\mathbf{C}}$ is the projection operator on the convex set $\mathbf{C}$. The gradient direction of this Moreau-envelope is:

$$\nabla f(\bar{\mathbf{y}}) = \tau^{-1} (\bar{\mathbf{y}} - \hat{\mathbf{y}})$$

By setting $K = \mathbf{J}(\mathbf{y}_k)$ and $\mathbf{b} = \mathbf{J}(\mathbf{y}_k)\mathbf{y}_k - \mathbf{r}(\mathbf{y}_k)$, we can use (4.14) together with a proper step-size backtracking to optimize the original problem in Equation (4.13) for convex constraints $\mathbf{C}$. More details about the Moreau-enveloped problem can be found in Section 2.10 and [26, p. 29 ff].

4.5.3 Benchmarks

Performance of the proposed optimization method was evaluated by solving a series of benchmark problems of increasing complexity. Fully characterized reference solutions $\hat{\phi}$ for ventricular activation sequences for a given domain Ω were generated by defining an orthotropic diffusion tensor $\hat{D}$ and initial activation $\hat{S}$ with initial timings $\mathbf{g}$. Data observed at the outer surface of the domain ($\Omega_\mathrm{O} \subset \Omega$) were used then as inputs for the inverse FIMIN-based optimization. Starting from various default assumptions we attempted to identify either the eigenvalues v_f, v_s and v_n of the velocity tensor D, as well as initiation timings $\phi(\mathbf{v}_0)$ on S, or both.

Two setups were considered. First, a simple computationally inexpensive 2D tissue sheet was generated that allowed for sufficiently short simulation cycles useful in identifying fundamental algorithmic issues and devising strategies of how to address these. Secondly, an anatomically accurate human biventricular (BiV) model with physiologically realistic activation sequences was utilized to evaluate the method's ability to identify the parameters governing the ventricular activation sequence from limited observations recorded from the epicardial surface.

4.5.3.1 2D tissue sheet

A simple 2D sheet of size $2 \times 2\text{cm}^2$ was discretized at 1mm resolution, yielding a small grid, which was triangulated using Delaunay triangulation into 722 triangles (see Figure 4.4 A)). In [33] it was shown that, using an eikonal model, a spatial resolution of 1mm is capable of capturing the most important features of wavefronts for anisotropy ratios found in cardiac applications. An arbitrary, but smooth, diffusion tensor field $\hat{D}$ with EAS at the tissue's center was used for generating a reference activation sequence $\hat{\phi}$ by solving Equation (4.11) using FIM for triangular domains. Data along the boundary of the domain Ω_{O} were used to identify $D(\mathbf{x})$ by optimizing according to Equation (4.11).

4.5.3.2 Human biventricular model

A human biventricular (BiV) anatomy model was reconstructed from an end diastolic 3D balanced steady state free precession (SSFP) cardiac magnetic resonance imaging acquisition in a sagittal orientation with whole-heart coverage and an isotropic resolution of 1.3 mm. Details on the model building process have been reported previously elsewhere [6, 39],but are briefly summarized here. A tetrahedral finite element mesh was generated [99] from a segmented image stack at an average resolution of 880μm (Figure 4.5 A, top panel). Fiber architecture was incorporated assuming a linear rotation of fiber angles from $-60°$ at the epicardium to $+60°$ at the endocardium using a rule-based method[9] (Figure 4.5 A, bottom panel). This resulted in a BiV model consisting of $\sim 5 \cdot 10^5$ nodes and $\sim 2.6 \cdot 10^6$ tetrahedral elements. To ease computational load, the original model was downsampled to an approximate resolution of about 1.3mm with $\sim 1.1 \cdot 10^5$ nodes and $\sim 5.8 \cdot 10^5$ tetrahedral elements.

Reference Solutions

The forward model for generating the ventricular activation sequence was based on the following considerations. Both anatomical [40, 92] and experimental [47] mapping studies that utilize *ex vivo* human hearts provide evidence that the electrical activation of the ventricles is initiated by the HPS [65] with EAS occuring at PVJs. Purkinje ventricular junctions (PVJs) within a healthy human LV can be approximated by a tri-fascicular conduction system [104] consisting of three major fascicles located high on the anterior paraseptal wall, $\mathbf{x}_{\text{LV,a}}$, in a central area at the septal endocardium, $\mathbf{x}_{\text{LV,s}}$, and in a posterior paraseptal area at about one third of the LV long axis above the apex, $\mathbf{x}_{\text{LV,p}}$. Similarly, in the RV fascicles were assumed to be located low on the the septal endocardium, $\mathbf{x}_{\text{RV,s}}$, at the base of the pulmonary trunk, $\mathbf{x}_{\text{LV,p}}$, and at an area high on the anterior wall near the junction of the right atrium, $\mathbf{x}_{\text{RV,a}}$. Spread of activation from EAS occurs at much higher CVs within the subendocardial layer than in the bulk myocardial wall due to the fast-conducting properties of the Purkinje network [45] or by a more abundant expression of sodium channels in subendocardial myocytes [101]. Tissue patches surrounding EAS are asummed to activate instantaneously.

Size, location, and timings of the EAS determining $\mathbf{S}$ and $\mathbf{g}$ are therefore key determinants shaping the activation sequence of the LV, as well as the prescribed orthotropic CVs within the domain. Thus, two different activation sequence models were used that both approximated the HPS with different degrees of fidelity: a topologically realistic model of the HPS with a larger number of PVJs (BiV-HPS), and a simpler trifascicular model (BiV-3F) comprising of only the three main fascicles in LV and RV. The BiV-HPS topolgy was obtained by starting with a manually delimited, scanner-based rabbit HPS and transferring it to the human mesh with universal ventricular coordinates[10]. Sizes and locations of EAS in $\mathbf{S}$ are indicated in Figure 4.5 B-C. The six distinct EAS, $\mathbf{x}_{\mathrm{LV,a}}$, $\mathbf{x}_{\mathrm{LV,s}}$, $\mathbf{x}_{\mathrm{LV,p}}$, $\mathbf{x}_{\mathrm{RV,a}}$, $\mathbf{x}_{\mathrm{RV,s}}$, $\mathbf{x}_{\mathrm{RV,p}}$, consisted of ~ 230 discrete nodes.

Reference solutions for BiV-HPS and BiV-3F model were generated using a set of default parameters that produce physiological activation patterns in line with measurements in humans [47]. In both cases, CVs were chosen homogeneously with $v_{\mathrm{f}} = 0.6\ m/s$, $v_{\mathrm{s}} = 0.4\ m/s$, $v_{\mathrm{n}} = 0.2\ m/s$, respectively, except for a subendocardial layer where a higher isotropic CV was prescribed with $v_{\mathrm{f}} = v_s = v_n = 1.5\ m/s$. In the BiV-HPS model, endocardial activation is governed by the topology of the HPS network, subendocardial and myocardial definitions of CV within the orthotropic diffusion tensor D, and transduction delays across PVJs when thin essentially 1D fibers of the HPS network initiate propagation in the large mass of the ventricular myocardium [17]. Initiation timings of PVJs therefore ranged from 0.58 ms up to 38.76 ms. Activation of the BiV-3F model was initiated by prescribing EAS timings, in the LV at $t_0 = 0\ ms$, 7 ms and 3 ms to $\mathbf{x}_{\mathrm{LV,s}}$, $\mathbf{x}_{\mathrm{LV,a}}$ and $\mathbf{x}_{\mathrm{LV,p}}$ fascicle, respectively, and in the RV at $t_0 = 1\ ms$, 9 ms and 15 ms to $\mathbf{x}_{\mathrm{RV,s}}$, $\mathbf{x}_{\mathrm{RV,a}}$ and $\boldsymbol{x}_{\mathrm{RV,p}}$ fascicle, respectively.

Optimization

Unlike for the 2D sheet, no attempts were made to identify eigenvectors and eigenvalues of D for all elements in the mesh as the high dimensionality of this problem would render the optimization prohibitively expensive. Rather, we subdivided the mesh into N partitions and identified in each partition i a set of CVs v_{f_i}, v_{s_i}, v_{n_i}. That is, the eigenvectors $\mathbf{f}_j, \mathbf{s}_j, \mathbf{n}_j$ of all D_j were considered given in each tetrahedral element j and optimization was performed only w.r.t. to the CVs in one partition i, i.e. the optimized CVs in a tetrahedron j v_{j,f_i}, v_{j,s_i}, v_{j,n_i} were identical for all tetrahedra in partition i. The velocity tensor D in tetrahedra j of partition i is given then as

$$D_j\left(v_{f_i}, v_{s_i}, v_{n_i}\right) = v_{f_i}^2\, \mathbf{f}_j \otimes \mathbf{f}_j + v_{s_i}^2\, \mathbf{s}_j \otimes \mathbf{s}_j + v_{n_i}^2\, \mathbf{n}_j \otimes \mathbf{n}_j$$

where D_j belongs exactly to one partition i. In addition, to limit the amount of anisotropy introduced during optimization, constraints were imposed on the admissible minimum and maximum CVs chosen as $v_{\min} = 0.10 m/s$ and $v_{\max} = 1.75 m/s$. This limited the maximum possible anisotropy ratio $\left(\frac{v_{\max}}{v_{\min}}\right)$ to ≤ 17.5. Additionally, the Huber regularization for this example is calculated on the conduction velocities, i.e. we regularize $\nabla v_{f/s/n_i}$. The optimization problem for the BiV model is therefore

$$\begin{aligned}
&\min_{v_{f_i}, v_{s_i}, v_{n_i}, \mathbf{g}} \frac{1}{2} \left|\left| \phi(v_{f_i}, v_{s_i}, v_{n_i}, \mathbf{g}) - \hat{\phi} \right|\right|^2_{2,\Omega_{\mathrm{O}}} + \frac{\gamma}{2} \sum_{j=1}^{|T|} \left|\left| \nabla D_j \left(v_{f_i}, v_{s_i}, v_{n_i}\right) \right|\right|_H \\
\text{s.t.} \quad & D \in S^n_{++}, \phi(\mathbf{x}) = \mathrm{FIM}(D(v_{f_i}, v_{s_i}, v_{n_i}), M, S(\mathbf{g}))(\mathbf{x}), \\
& v_{\min} < v_i < v_{\max}
\end{aligned} \tag{4.15}$$

The epicardial surface Ω_{O} of the BiV model was spanned by $\approx 1.4 \cdot 10^4$ discrete nodes. Optimization of the initiation timings $\phi(\mathbf{x}_0)$ of PVJs in the BiV-3F was performed for each discrete node separately.

The partitioning between BiV-3F and BiV-HPS models differed. In the BiV-3F model only $N = 2$ partitions were used, the endocardium and the remainder of the heart, whereas in the BiV-HPS model $N = 100$ partitions of equal size were used that were randomly located throughout the domain Ω. Partitioning in the BiV-HPS case was performed by converting the tetrahedral mesh into an unweighted dual-graph, where an edge is equal to a face connecting two tetrahedra, and applying then the min-edge-cut algorithm as implemented in the METIS library [76].

4.5.3.3 Robustness

Real world measurements are limited in terms of spatio-temporal resolution and afflicted by uncertainties. With regard to measuring epicardial activation patterns, spatial location of recording sites, electrogram fractionation and associated difficulties in reliably deriving a marker representing the instant of local activation, and noise are the main sources of error. To probe the robustness of the FIMIN algorithm against these errors, the BiV benchmarks were repeated in presence of measurement errors and with undersampled data. Original reference data $\hat{\phi}$ were perturbed in two different ways:

1. Undersampled data were generated by selecting a subset of observed points on Ω_{O}, $\hat{\Omega}_{\mathrm{O}} \subset \Omega_{\mathrm{O}}$. The remainder of points on Ω_{O} were generated from $\hat{\Omega}_{\mathrm{O}}$ by inverse distance weighting.

2. Measurement errors were introduced by perturbing observed data by adding a normal distributed random noise with zero mean and σ standard deviation:

$$\forall \mathbf{v} \in \hat{\Omega}_{\mathrm{O}} : \tilde{\phi}(\mathbf{v}) = \hat{\phi}(\mathbf{v}) + \mathcal{N}(0, \sigma) . \tag{4.16}$$

where σ was chosen at the order of typically used temporal sampling intervals up to 8 ms.

4.5.4 Results

4.5.4.1 2D-Sheet

To demonstrate the general applicability of the FIMIN algorithm, we allowed the maximum degree of freedom by optimizing each element of every D seperately with no constraints other than D being required to be a metric (see Equation (4.11)). Data at the observation domain were generated by initiating a wavefront at the tissue center with D defined by the eigenvectors illustrated in Figure 4.4 A and a homogeneous set of CVs v_f and v_s given as 0.6 and 0.34 m/s. To provide a more realistic and continuous scenario, D was slightly smoothed to ease optimization. Additionally, the 2D experiments also considered the eigenaxes of D to be unknown and optimizable. We optimized D by using the FIMIN Algorithm 4.3 and completely random initial CVs within the range $v_{f,0}, v_{s,0} \in [0.15, 0.98]\, m/s$. The initial random D and the corresponding initial activation sequence without optimization is shown in Figure 4.4 B. Optimization with the FIMIN, without any regularization ($\gamma = 0$) yielded an optimal, zero energy solution, but with a highly anisotropic and non-smooth D (Figure 4.4 C). This can be attributed to the fact that FIM gradients track the wavefront and compute the influence of changing D on each vertex of the domain along the geodasic paths from an initial starting point. Since the gradient is propagated along the geodesic paths and there is no cost associated with errors outside Ω_{O}, many geodesic paths collapsed into a single path. Diffusion tensors that did not fall on any geodesic path were not updated as no gradients were propagated across them. Such highly anisotropic and non-smooth Ds are a rather unlikely solution to the problem, even though they may perfectly approximate the target function $\hat{\phi}$ on Ω_{O}.

As shown in Figure 4.4 D, already with a low value of $\gamma = 5 \cdot 10^{-3}$ the resulting D provides a much smoother solution closer to the original diffusion tensor field (compare Figure 4.4 C and D).

The 2D toy example demonstrated that the ill-posedness of the problem could not be overcome without a good regularization of the objective function. While still not perfect, the regularized solution in Figure 4.4D provides a less complex and much more likely D than the unregularized optimal solution in Figure 4.4C and demonstrates the viability of our approach. We tried two other, probably more physiological plausible, 2D experiments that can be seen in Section A.1, Figures A.11 and A.12.

4.5.4.2 Biventricular Models

In the BiV anatomy model with a given fiber and sheet architecture (Figure 4.5A) reference activation sequences were generated with the BiV-3F and BiV-HPS configuration (see Figure 4.5B-C), respectively. Using FIMIN we attempted to identify the timings $\phi(\mathbf{x}_0)$ and velocities v_{f}, v_{s} and v_{n} in each of the N partitions of the myocardium from measurements of ϕ, taken at the epicardial surface Ω_0 only. Eigenvectors of the D, $\mathbf{f}(\mathbf{x})$, $\mathbf{s}(\mathbf{x})$ and $\mathbf{n}(\mathbf{x})$, were considered given.

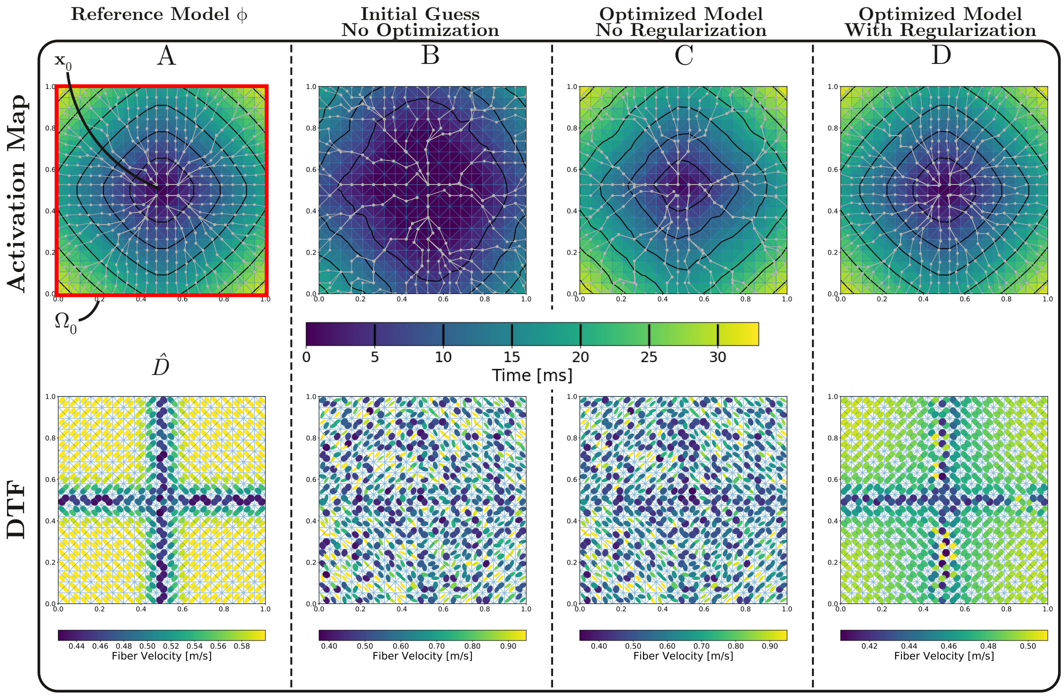


Figure 4.4: A) 2D benchmark setup showing domain Ω, EAS $\mathbf{x}_0$ in the sheet's center, the reference activation map $\tilde{\phi}$, diffusion tensor field $\hat{D}(\mathbf{x})$ and the measurement domain Ω_0. Note that the velocity colormap is slightly different for each of the diffusion tensor field due to high variation of fiber velocities between experiments. B) Initial solution before optimization. C) Solution for optimizing Equation (4.11) with no regularization ($\gamma = 0$). The SSE of activation times is practically 0, but the resulting D is highly anisotropic and non-smooth. Many geodesic paths (grey lines) collapse into one. D) Solution for optimizing Equation (4.11) with $\gamma = 5 \cdot 10^{-3}$. The energy is still very low, but the regularization yields a smoother D and avoids high anisotropies. Geodesic paths are spread over more points and the contour lines are very close to the desired solution.

FIMIN was initialized assuming that activation is initiated at all EAS at the same time, i.e. $\phi(\mathbf{x}_0) = 0\,ms$ was chosen. The initial velocities for all partitions were set to 0.93 m/s, 0.66 m/s and 0.54 m/s for $v_{\mathrm{f},i}$, $v_{\mathrm{s},i}$ and $v_{\mathrm{n},i}$ respectively. Note that these initial velocities differed significantly from those used for computing the reference solution in terms of both magnitudes as well as the ratios between them. Further, no distinction was made between the orthotropic myocardium (0.6, 0.4 and 0.2 m/s for v_f, v_s and v_n, respectively) and the much faster conducting isotropic subendocardial layer (1.5 m/s).

Optimizing Equation (4.15) w.r.t. velocities $v_{\mathrm{f},i}$, $v_{\mathrm{s},i}$ and $v_{\mathrm{n},i}$ and EAS timings $\phi(\mathbf{x}_0)$ proved to successfully reduce the error on both epicardium Ω_0 and myocardium Ω. A

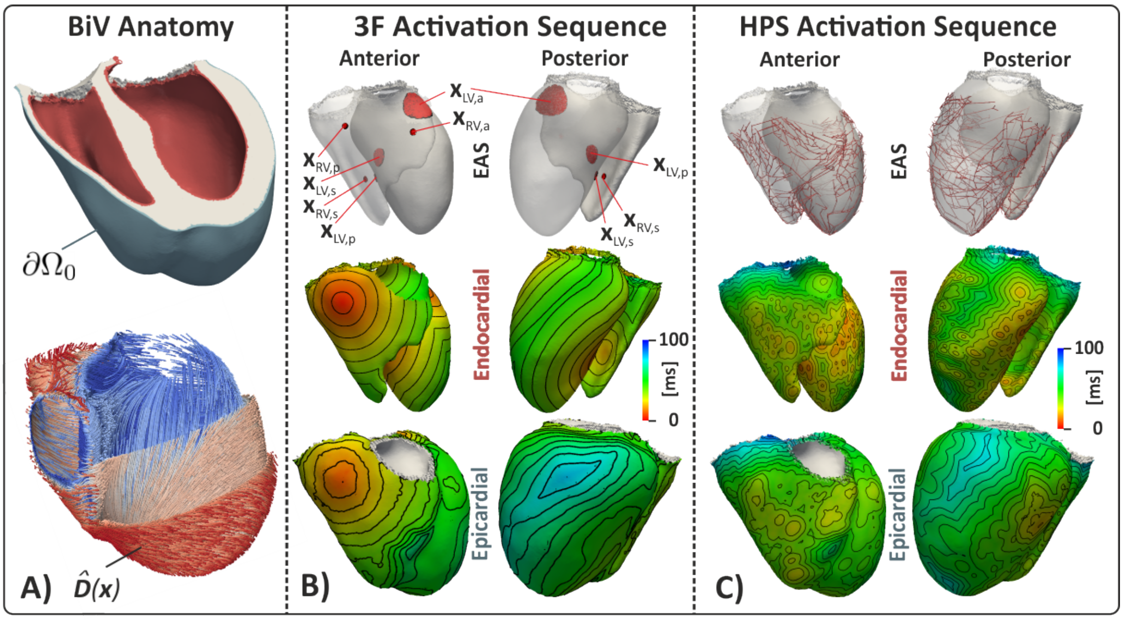


Figure 4.5: Biventricular benchmark setups showing domains Ω, earliest activation sites $\mathbf{x}_0$ (EAS), diffusion tensor field $\hat{D}(\mathbf{x})$ and the domain boundary Ω_0 from which observed data are recorded. A) Biventricular anatomy highlighting endocardial and epicardial surface (top panel) and fiber arrangement (bottom panel). B) and C) show trifascicular (3F) and HPS initiated activation sequence, driven by EAS activating at prescribed $\phi(\mathbf{x}_0)$ (top panels).

comparison of activation isochrones of initial and optimized model to the reference model is shown in Figure 4.6 for both 3F and HPS model. The activation sequence $\phi_{\text{HPS}}(\mathbf{x})$ is significantly more complex than $\phi_{\text{3F}}(\mathbf{x})$. Owing to the larger number and the smaller sizes of EAS's in the HPS model wave fronts amalgamated inside the myocardial wall well before breaking through at the epicardium. Thus, the number of epicardial breakthrough sites was much smaller than the number of endocardial initiation sites, as evident by comparing endocardial and epicardial isochrone patterns in the HPS case (see Figure 4.6). This rendered the estimation of timings $\phi(\mathbf{x}_0)$ and velocities $v_{\text{f},i}$, $v_{\text{s},i}$ and $v_{\text{n},i}$ a challenging task. Similar to the 2D example in Section 4.5.4.1, the proper choice of γ in Equation (4.11) played an equally important role in keeping the optimization from overfitting onto the epicardial data Ω_{O}.

In Figure 4.7, the comparison of the isochrones of our optimized model (black), versus the reference model (white) at a fixed time is shown, for both the 3F and HPS model and the epi-, as well as the endocardium. The overlap of both isochrones on the epicardium show that FIMIN is able to effectively adjust the conduction velocity parameters to minimize the distance of the activation times of the model and reference on the epicardium to a high - and on the endocardium to a lesser, but still significant - degree. The online version of the paper also includes a video, showing this comparison over time using the initial, as well as the optimized model, accessible by clicking on either of the figures in Figure 4.7.

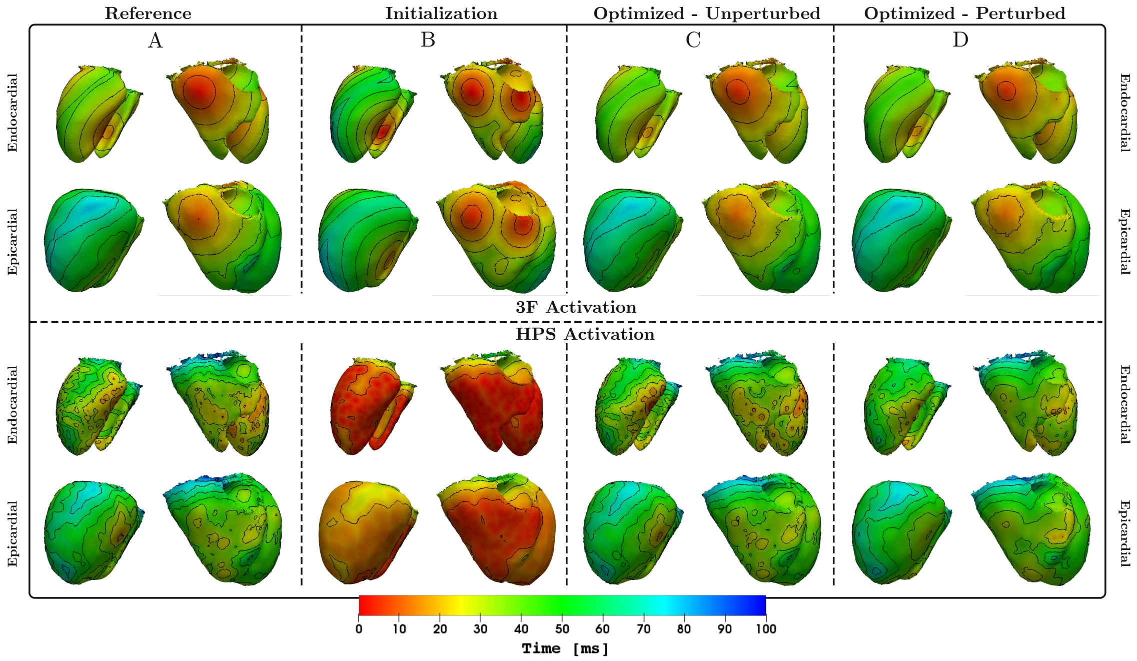
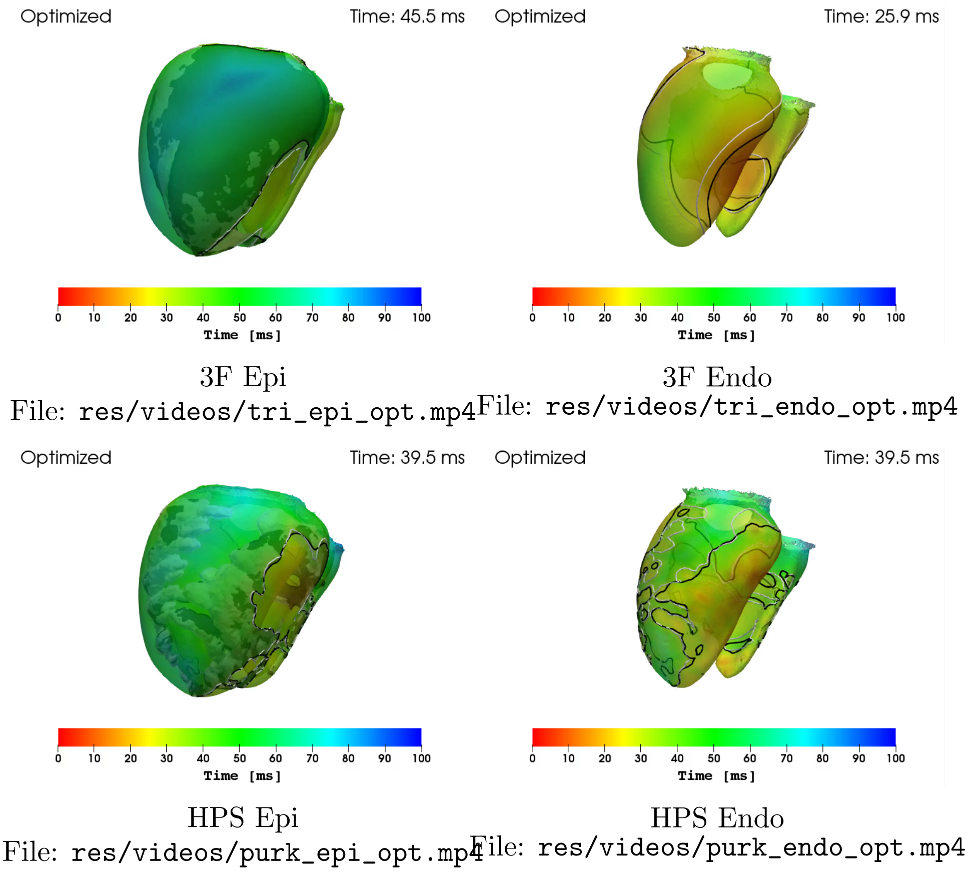


Figure 4.6: Top panels: Shown are endocardial and epicardial activation isochrones of ϕ_{3F} from a posterior and anterior viewpoint for the reference $\hat{\phi}_{3F}$ (A), the solution ϕ_{3F} using initial (B) and optimized (C) parameters. Additionally we show the result when optimizing against $\tilde{\phi}_{3F}$ from Equation (4.16) with the strongest tested perturbation (D) (see Section 4.5.4.3, Table 4.1, bottom right cell). Bottom panels: The same comparison is shown for ϕ_{HPS}.

The effect of optimization on reducing the variation of errors $\Delta_\phi = \hat{\phi} - \phi(\mathbf{y})$ is shown in Figure 4.8, considering only Δ_ϕ in the range $[-20, 20]$ ms. After optimization, errors on Ω_O were significantly reduced, following a normal distribution with $\mu_\Delta = -1.3ms$, $\sigma_\Delta = \pm 2.6\,ms$ and $\mu_\Delta = 0.7\,ms$, $\sigma_\Delta = \pm 3.7\,ms$ for 3F and HPS, respectively. Over the domain Ω, hidden to the optimization function, the overall behavior was comparable, with errors also converging close to a zero mean, albeit with a slightly skewed distribution. These results suggest that the hidden SSE function can be optimized over the whole ventricular domain Ω using data observed on the epicardium Ω_O only.

Optimization results with regard to conduction velocities are shown in Figure 4.8. For the less complex activation sequence ϕ_{3F} where Ω was partitioned into two velocity domains only – myocardium and subendocardial layer – the optimized CV triplets of 0.68/0.29/0.18 m/s and 1.69/1.68/1.25 m/s were close to the true values of 0.6/0.4/0.2 m/s and 1.5/1.5/1.5 m/s (compare Section 4.5.4.2). In contrast, for the more complex activation sequence ϕ_{HPS}, where 100 partitions were used, a higher variability in optimized conduction velocities was witnessed, as shown in Figure 4.9. The optimized velocity dis-



3F Epi
File: `res/videos/tri_epi_opt.mp4`

3F Endo
File: `res/videos/tri_endo_opt.mp4`

HPS Epi
File: `res/videos/purk_epi_opt.mp4`

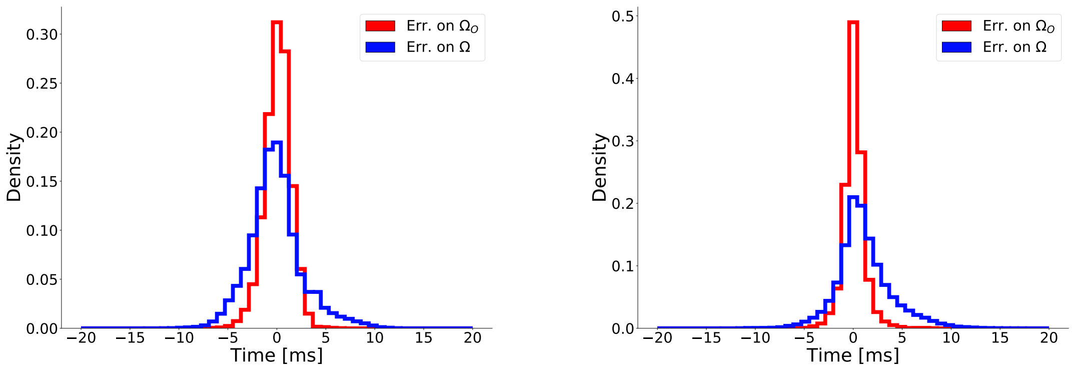
HPS Endo
File: `res/videos/purk_endo_opt.mp4`

Figure 4.7: Comparison of the isochrones at a certain time for the optimized model (black), versus the reference model (white). Clicking the images will open a video showing the comparisons over time of the initial and the optimized model (online version only).

tributions were centered around the reference velocities, but outliers existed, particularly in the distribution of v_f.

A comparison between true and optimized EAS timings $\phi(\mathbf{x}_0)$ is given in Figure 4.10. For both sequences ϕ_{3F} and ϕ_{HPS} true and optimized EAS timings largely overlap, but the distribution of optimized timings is more spread out which is a consequence of optimizing each discrete node of the EAS regions independently. Moreover, an apparent bias towards later timings is observed which was particularly pronounced for the ϕ_{3F} sequence.

The observed spread in both CVs and EAS timings could be mitigated by increasing the value of γ in Equation (4.15), leading to a more narrow velocity distribution. However, errors in ϕ at the epicardium Ω_{O} were compensated by the optimization by shifting the initiation timings $\phi(\mathbf{x}_0)$ which were not subjected to regularization (data not shown).



3F HPS

Figure 4.8: Effect of optimization on the distributions of errors Δ_ϕ in (a) 3F and (b) HPS model. Blue traces show Δ_ϕ on the epicardium Ω_O only, whereas red traces consider the whole domain Ω. The error distributions on Ω can be very well approximated by a normal distribution, albeit slightly skewed. Means and standard deviations were (a) $\mu_\Delta = -1.3ms$, $\sigma_\Delta = \pm 2.6ms$ and (b) $\mu_\Delta = 0.7ms$, $\sigma_\Delta = \pm 3.7ms$ for 3F and HPS model, respectively.

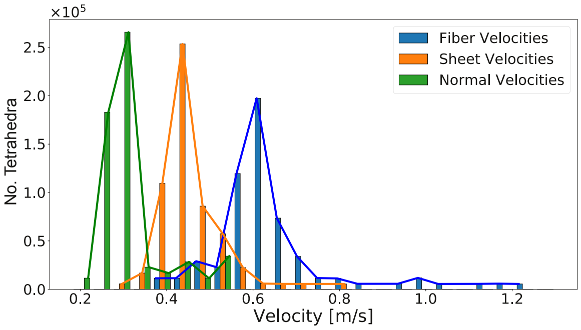
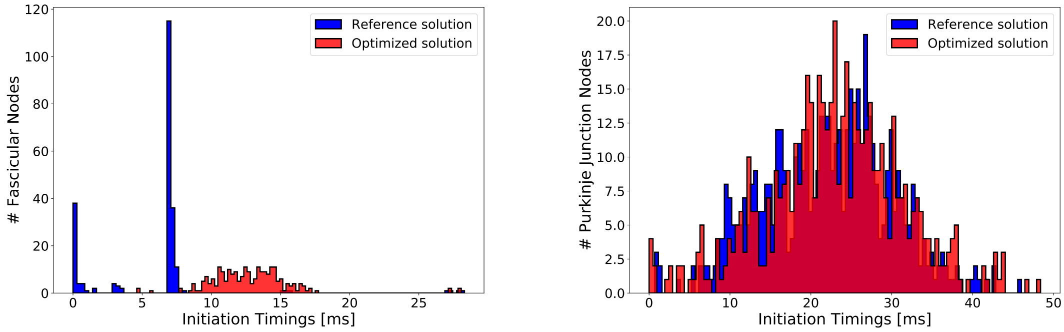


Figure 4.9: Distribution of optimized velocities v_f, v_s and v_n of the HPS model. The bars represent the actual, narrow distributions, which can be attributed to splitting the mesh into 100 partitions. The lines show the envelope of the corresponding bars with the same color. Reference velocities were $0.6/0.4/0.2\frac{\text{m}}{\text{s}}$ for v_f, v_s and v_n, respectively.

4.5.4.3 Robustness

To gauge the potential of the FIMIN algorithm under real world conditions, input data on Ω_O were perturbed by reducing spatial sampling and adding noise, as detailed in Section 4.5.3.3. A qualitative comparison in terms of activation isochrones between BiV-3F and BiV-HPS models with maximum perturbation ($\frac{|\hat{\Omega}_\text{O}|}{|\hat{\Omega}|} = 0.7\%, \sigma = 8\,ms$) relative to the reference and unperturbed optimization is also shown in Figure 4.6. While minor



3F

HPS

Figure 4.10: Distribution of reference $\hat{\phi}(\mathbf{x}_0)$ in blue versus opimized $\phi(\mathbf{x}_0)$ EAS timings in red.

deviations were visible in the BiV-3F case, there was a high level of agreement in isochronal patterns suggesting that the overall structure of the BiV-3F model was captured despite coarse spatial sampling and high noise levels.

In the BiV-HPS case, discrepancies were more pronounced as details of the fine-grained HPS-mediated break-through patterns were lost. Nonetheless, even under these severely perturbed conditions the model approximated the overall epicardial activation patterns with sufficient accuracy for real world applications.

As shown in Figure 4.8 for the noise-free model, the optimization reduces the errors of activation times and usually converges to a normal distribution with near-zero mean. Noisy and undersampled input data followed a similiar normal distribution of error (data not shown) as shown in the noise-free model Figure 4.8. Deterioration due to noise and undersampling, can therefore be characterized by computing mean and standard deviation of total errors relative to the known reference solution. Results for all scenarios considered are summarized in Table 4.1. As evident from inspection of data in Table 4.1, relative to the optimal case in absence of noise and undersampling (upper left corner in Table 4.1 with $\hat{\Omega}_\text{O} = \Omega_\text{O}$ and $\sigma = 0\,ms$), errors increased with increasing noise and decreasing spatial sampling, but the chosen regularization kept the FIMIN algorithm from overfitting to the epicardial activation times.

4.5.5 Discussion

This study reports on the development of a novel FIMIN algorithm for solving an inverse eikonal problem. Conceptually, the method attempts to identify the key factors governing the ventricular activation sequence – location and timing of EAS and the velocity tensor field D – from sparse and noisy activation data $\hat{\phi}$ sampled from the epicardial surface.

The overall feasibility and limitations of the method were investigated *in silico*. An anatomically accurate human BiV model was used to generate high fidelity reference ac-

	3F-Model			HPS-Model		
σ $\lvert\hat{\Omega}_{\mathrm{O}}\rvert$	0 ms	4 ms	8 ms	0 ms	4 ms	8 ms
$\hat{\Omega}_{\mathrm{O}} = \Omega_{\mathrm{O}}$	-0.1 ± 2.8	1.7 ± 3.0	0.4 ± 4.2	1.0 ± 3.5	1.5 ± 3.6	0.5 ± 3.6
$1 \cdot 10^3$	-1.2 ± 3.7	-0.3 ± 2.8	-0.9 ± 3.8	4.0 ± 5.4	3.5 ± 5.4	1.8 ± 6.4
$1 \cdot 10^2$	1.2 ± 3.9	-1.55 ± 4.4	-1.2 ± 3.5	5.3 ± 6.9	4.8 ± 6.5	2.1 ± 7.2

Table 4.1: Shown are the means and standard deviations of total errors Δ_ϕ in ms over the whole domain Ω between reference and optimized solutions for varying degrees of noise σ and spatial undersampling $\hat{\Omega}_{\mathrm{O}}$. Initial errors of 3F and HPS model were $\mu_\Delta = -1.4$ms, $\sigma_\Delta = \pm 11.3$ms and $\mu_\Delta = -23.0$ms, $\sigma_\Delta = \pm 8.3$ms for 3F and HPS model, respectively.

tivation sequences $\phi(\mathbf{x})$ as ground truth solutions – either the simplified ϕ_{3F} sequence initatiated by a trifascicular HPS, or, the ϕ_{HPS} sequence driven by a topologically more realistic HPS network – by solving the anisotropic eikonal equation. Using activation times sampled from the forward solution over the epicardial surface and assuming the location of EAS's were known, the corresponding inverse problem was solved using the FIMIN algorithm to identify the governing input parameters of the forward model that optimally fit the epicardial activation map.

Using the epicardial activation map at full resolution we first demonstrated that a reconstruction of activation sequence in 3D is, in principle, feasible under the assumption of known location of EAS. The robustness of the method under closer to real world conditions was evaluated by reducing the spatial resolution of the observed epicardial activation map and by adding noise. Our results suggest that the FIMIN algorithm is able to recover the full 3D activation sequence even under these more realistic conditions with spatial resolutions that are achievable with currently used clinical mapping systems. Contrasting the uncertainties in clinical data with the accuracy achieved in reconstructing $\phi(\mathbf{x})$ suggests that the FIMIN method may be suitable for the patient-specific parameterization of activation models in future clinical applications.

4.5.6 Identifying the ventricular activation sequence

The identification of the ventricular activation sequence $\phi(\mathbf{x})$ from measurements taken from the body surface – as in a standard inverse electrocardiographic imaging problem – or from the outer epicardial surface of the ventricles – as suggested in this study – constitutes a severely ill-posed problem. Any attempt to address this kind of problem depends therefore critically on the use of *a-priori* knowledge to constrain the problem and appropriate regularization techniques to enforce the constraints. Conceptually, the ventricular activation sequence $\phi(\mathbf{x})$ is governed by three factors only, the location of EAS, their timings, $\phi(\mathbf{x}_0)$, and the conduction properties of the tissue as encoded in the velocity tensor field $D(\mathbf{x})$. This problem, in its most general interpretation, can be considered extremely high dimensional. In this study, three simplifying assumptions were made to reduce the dimensionality of the problem and to keep the FIMIN-based optimization tractable: i) the location of EAS's were assumed to be known; ii) the eigenaxes of $D(\mathbf{x})$ were prescribed on a per-rule basis [9] informed by anatomical studies [47], only the tensor eigenvalues representing anisotropic velocities were considered unknown; and, iii) the spatial variation of the eigenvalues of $D(\mathbf{x})$ was regularized by partitioning the ventricles into N domains and enforcing velocity triplets $v_{\mathrm{f},i}$, $v_{\mathrm{s},i}$ and $v_{\mathrm{n},i}$ to be constant throughout a partition i.

Under these assumptions FIMIN was able to find parameter sets comprising $\phi(\mathbf{x}_0)$ and $v_{\mathrm{f},i}$, $v_{\mathrm{s},i}$ and $v_{\mathrm{n},i}$ which approximated $\phi(\mathbf{x})$ quite well at the epicardial surface Ω_{O} where activation maps predicted by the FIMIN model ϕ were compared to observations $\hat{\phi}$, but also over the entire myocardial volume Ω where data on $\hat{\phi}$ were hidden to the optimization.

In general, results tended to be more accurate for the ϕ_{3F} than for the ϕ_{HPS} sequence due to the more direct relationship between EAS and epicardial breakthrough sites, as well as the lower complexity of the optimization model. In the ϕ_{HPS} case there was a larger number of EAS's in each ventricle. These initiated numerous wavefronts at the endocardium which merged within the ventricular myocardium, thus yielding a reduced number of epicardial breakthrough sites.

A notable deviation of optimized CVs from the reference CVs was wittnessed in the BiV-HPS case which can be attibuted to a number of factors. First, the high number of EAS's is likely to lead to a multimodal problem where the errors Δ_ϕ can be compensated by both changes in initiation timings as well as changes of the CVs. Second, the gradient of velocities is not uniformly distributed over the domain, favoring optimization of the partitions closer to the epicardium as their gradient tends to have a higher magnitude. The Huber regularization mitigates this behavior up to a certain degree, but increasing γ amplifies the first mentioned problem, favoring the optimization of initiation timings over conduction velocities. Partitioning the mesh into regions of constant conduction velocities significantly reduced the computation load, but the regularity of conduction velocities across the whole domain is more effectively driven by our choice of γ.

This activation time which was earlier relative to the actual PVJ activation times in the reference model (see Figure 4.10). Thus the initial activation sequence leads the reference sequence throughout the entire domain Ω, i.e. $\Delta_\phi < 0$. Both optimized BiV-3F and BiV-HPS models yielded a normal distribution of errors on Ω_{O}, which is a common effect when optimizing using the SSE function. In the case of not perfectly optimizable functions, the SSE function will lead to a normal distribution of errors with zero mean, as can be seen in Figure 4.8.

Finding the correct velocities by optimization was complicated by the fact that initiation timings $\phi(\mathbf{x}_0)$ were also assumed to be unknown. This biased the optimized solution towards later initiation timings $\phi(\mathbf{x}_0)$ (see Figure 4.10) which were compensated by higher conduction velocities (see Figure 4.9).

4.5.7 Robustness

In view of potential applications of FIMIN in an experimental or clinical context, the robustness with regard to spatial resolution and distortions due to noise is of pivotal importance as current modalities for measuring $\phi(\mathbf{x})$ suffer from a number of limitations. With regard to spatial undersampling and noise FIMIN-based reconstructions of $\phi(\mathbf{x})$ were shown to be surprisingly robust. Reconstructed isochronal patterns appeared low-pass filtered when compared to unperturbed reconstructions, but the overall structure of the solution was retained, even under the most severe conditions where only 100 epicardial data points were used and the noise added to arrival times was normally distributed with a standard deviation of 8ms. The undersampling corresponded to approximately 0.7% of the available epicardial points, with an average spacing of 14.2mm between the measurement points.

In general, it can be observed from Table 4.1 that the final error after the optimization is mostly dependant on the undersampling (vertical) rather than the noise added to the observed data (horizontal). This can be attributed mainly to our cost function: We chose a mean-squared-error function in our problem formulation (see Equation (4.11)), which is an appropiate assumption to mitigate the effects of Gaussian noise. Although mean and standard deviation vary between most of the experiments in horizontal direction of Table 4.1, the final mean-squared error is very similar for all of these experiments.

Robustness of FIMIN-based reconstructions of the activation sequence must be viewed in the context of data uncertainty of measured activation maps. In general, a highly accurate observation of $\phi(\mathbf{x})$ at the organ scale throughout Ω is not feasible with currently available technology. Owing to the physics of propagating depolarization wavefronts which is governed by fast transients ($<$ 1ms) that translate into steep wavefronts ($<$ 1mm), mapping technologies for an accurate registration of $\phi(\mathbf{x})$ would be required that offered the ability to measure wavefront arrival times with sub-millimeter and sub-millisecond resolution throughout the entire ventricular myocardium. Such measurements are massively invasive and only applicable in an experimental setting, but even the most advanced mapping techniques do not meet these requirements. Optical mapping techniques provide better spatial resolution, at least in theory, but are affected by a number of significant artifacts related to signal distortion due to integration effects mediated by photon scattering [15]. Further, optical measurements are confined to superficial tissue layers as they are not able to record, in a reliable fashion, any information from the depth of the tissue. While panoramic imaging systems exist, the typical field of view of mapping systems is limited and cannot cover the entire epicardial surface Ω_O. Finally, while cardiac tissue is often considered a functional syncytium, it is a discrete structure which inevitably results in wavefront fractionation, even in perfectly healthy tissue. These deviations from a continuum are reflected in signal fractionation which often renders the determination of arrival times ambiguous.

4.5.8 Relation to inverse ECG imaging problems

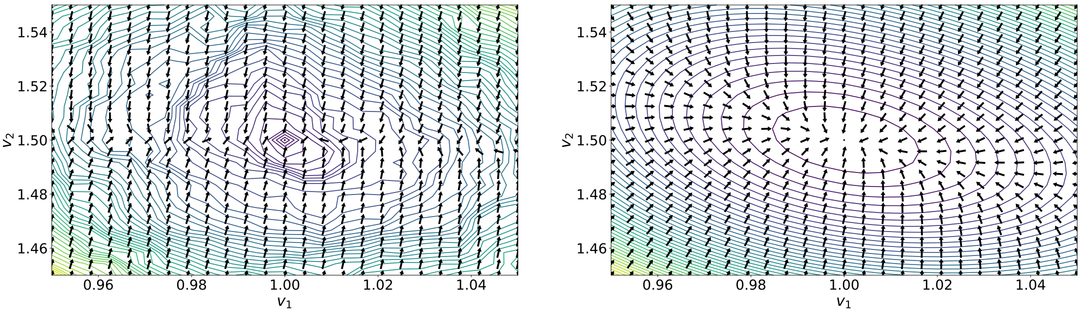
The FIMIN method for relating epicardial arrival times to the ventricular activation sequence is a non-standard ECGi problem. Instandard ECGi problems electrical potentials in the form of electrocardiograms or body surface potential maps are used as observations to infer different aspects of the ventricular activation sequence. In physical terms, inverse ECGi yields information on transmembrane voltages [68, 124], extracellular potentials [11, 108] or local activation/recovery times [48, 67, 72, 120]. which are, typically, reconstructed on the epicardial surface, but methods for endocardial or transmural reconstructions also exist. That is, the observations used as input in this study are often considered the solution in ECGi problems. As such the FIMIN is not directly applicable to ECGi problems without further extensions as the FIMIN, in its current form, accepts activation times from the epicardial surface as inputs, but not electrical potentials. However, coupling of FIM-based activation models with forward predictions of electrical potentials at the body surface can be done very efficiently, as has been shown by us [87] and others [95, 128]. In particular, eikonal solvers in combination with appropriate source models and a lead-field approach [54, 97] as used in [95] can be solved in a computationally highly efficient manner. Such methods yield body surface potentials which are essentially equivalent to ground truth solutions computed with an expensive fullblown bidomain formulation [97]. To construct an ECGi method based on the FIMIN one could use a FIM-based forward model [87, 95] to compute transmembrane voltage maps throughout the myocardium, $V_m(\mathbf{x}, t)$. In the simplest case a fixed action potential shape $U(t)$ may be used and shifted by the computed activation times $\phi(\mathbf{x})$ to compute the source term $V_m(\mathbf{x}, t) = U(t - \phi(\mathbf{x}))$. This extension will be used later in Section 4.7.2.2 to solve the inverse ECG problem and could be employed here in a similar fashion.

4.5.9 Importance of ε

Initial experiments with the 2D sheet model demonstrated that appropriately choosing ε for Algorithm 4.1 is of pivotal importance. This can be exemplified in a simple perturbation experiment with the BiV model from Section 4.5.3.2, with a single CV triplet (v_f, v_s, v_n) for the whole domain. Increasing ε decreased the runtime to compute the FIM solution, but the resulting surfaces of the inverse optimization problem and the obtained gradients were significantly less smooth as compared to using a smaller ε (see Figure 4.11). In this example, only a weakly anisotropic D was used with the optimal choice of CVs being $v_f = v_1 = 1$ and $v_s = v_2 = 1.5$. It is worth noting that this effect was amplified by increasing anisotropy ratio $\frac{v_1}{v_2}$ (not shown).

4.5.10 Computational Costs

Computational costs of the FIMIN algorithm were significant. For instance, finding the optimal parameters in the BiV-HPS model last for ≈ 16 hours. The BiV models consisted



Energy Landscape with $\varepsilon = 10^{-3}$ Energy Landscape with $\varepsilon = 10^{-7}$

Figure 4.11: Energy landscapes for different values of ε in the FIM Algorithm. Contour lines visualize the cost function from Equation (4.11) we seek to minimize, with the global minimum being located at $v_1 = 1$ and $v_2 = 1.5$. Arrows indicate the normalized gradient descent direction obtained by using Algorithm 4.2. Smaller ε yields a smoother energy landscape which benefits optimization.

of $\sim 1.1 \cdot 10^5$ nodes and $\sim 5.7 \cdot 10^5$ tetrahedral elements. The FIM and the computation of the Jacobian in Algorithm 4.2 were implemented in C++, while the remainder of the optimization used Python/Numpy. As a stopping criterion for the FIMIN a maximum number of iterations of 1000 or a slowdown in the rate of convergence was used. Prolonged execution times could be attributed to the increasing anisotropy of the optimized model as well as to the small value of ε that was needed for the FIM algorithm to efficiently optimize the FIM function. The two major contributors to overall execution time were the FIM algorithm and the Jacobian computation.

Increasing ε, lowering the maximum number of iterations or further decreasing spatial resolution are viable ways to achieve shorter execution cycles. However, these measures are likely to increase errors due to lower precision or premature convergence. Constraining the maximum permissible anisotropy by regularization could also be beneficial, but this has not been investigated in this study. Finally, further speedups may be gained from optimized implementations. For instance, the computationally expensive parts of the FIM algorithm are well suited for a SIMD-architectures such as a GPU, albeit the size of the model and, consequently, its Jacobian would require a GPU with a large memory or a multi-GPU implementation.

4.5.11 Limitations

Owing to the severe degree of ill-posedness of the problem, a number of simplifying modeling assumptions were made that may restrict the applicability of FIMIN to real world clinical problems. First and most importantly, epicardial activation maps were used as inputs. While the acquisition of such maps is clinically feasible with minimally invasive techniques, such procedures are non-standard. To be of wider utility, non-invasive input data such potential maps from the body surface should be considered within a more standard inverse ECG imaging setting [30]. Secondly, the location of the various EAS were assumed to be known. There is currently no straight-forward way, in which the location of the EAS could be integrated as an optimizable parameter into the optimization problem in Equation (4.11), using only epicardial data. While endocardial mapping studies to reveal EAS are performed routinely in the clinic, the acquisition of endocardial maps relies on an invasive procedure. From an applied point of view, the location of EAS should therefore be considered unknown and must be identified with FIMIN as part of the optimization procedure. Clinically acquired endocardial activation maps are valuable for validation, but should not be used for inferring location of EAS *a priori*.

Another potential drawback is the choice of the number of partitions in which the conduction velocity is kept constant for the 3D-cases. This reduces complexity and therefore expressibility of the model. To model local heterogeneity of conduction velocities, like found in scarred regions, current methods utilize technologies relying on image-based modalities, e.g. [3]. If these are available, the partitions could be adapted to include the scarred regions in single partitions.

In the absence of this data, the partition size would need to be chosen smaller than the region of interest, to have a chance of finding the regions by the model alone. Excessively chosen partitions will not be able to express the desired local heterogeneity of conduction velocities, but rather average the found conduction velocity over the area/volume. In contrast, small partition sizes will lead to longer computational times, more needed optimization iterations and prohibitively large Jacobian-matrices for the Gauss-Newton updates. While the latter, memory drawback can be mitigated by using sparse matrix formats, the two former problems can only be addressed by by efficient parallelization, either on a cluster, or an efficient GPU implementation.

4.6 Endocardial contact mapping

In Section 4.5, we considered the heart domain Ω to be a subset of $\mathbb{R}^3$ with positive Lebesgue measure. In the following two works in Sections 4.6.2 and 4.6.3 [60, 61], we consider the problem of EAMs, created during catheter ablation therapy. Catheter ablation therapy is a common procedure to treat many arrythmias of the atria, such as AF. In most cases of ablation, a surgeon inserts a catheter in the patient's upper thigh into the inferior vena cava, to gain access to the RA. As there is no visual contact with the catheter, this is a fully digitally guided procedure, sometimes assisted with Computer Tomography (CT). Modern tools not only allow the localization of the catheter, but will also progressively build a model of the atria – the EAM. Additionally, when in contact with the endocardial atrial wall, the probe will measure electrical activation to generate an activation map projected onto the EAM. visualizes the rough outline of the procedure.

The target is now to identify electrical pathways of high, or low conductivity, which can result in AF. From a modeling point of view, the generated mesh represents a manifold $\mathcal{M} \subset \mathbb{R}^3$ to which the electrical propagation is bound. This means, we use a slightly altered version of (2.23), which is

$$\begin{cases} ||\nabla_{\mathcal{M}}\phi||_D = \sqrt{\langle D\nabla_{\mathcal{M}}\phi, \nabla_{\mathcal{M}}\phi\rangle} = 1 & \text{in } \Omega \\ \phi = g & \text{on } \Gamma \subset \partial\Omega \end{cases} \tag{4.17}$$

The theory behind the computation of $\nabla_{\mathcal{M}}$ was outlined in Section 2.7 . Note that by choosing D to have zero velocity in the direction normal to the manifold, we get the equality

$$\begin{aligned} \sqrt{D}\nabla_{\mathcal{M}}\phi &= \sqrt{D}\nabla\phi \\ \Rightarrow ||\nabla_{\mathcal{M}}\phi||_D &= ||\nabla\phi||_D \end{aligned} \tag{4.18}$$

which we will use in Section 4.6.3. The following two works in Sections 4.6.2 and 4.6.3 will use this formulation and problem setup to solve the inverse problem of identifying D from sparsely given ϕ. A variant of the FIM to solve eikonal on triangulated surfaces was presented in [52], which was used to generate the ground-truth in-silico models. Note that the tensor $D \in \bar{S}^3_{++}$ in the problem (4.17) is 3-dimensional. We employed for both of the mentioned works two modifications to the original problem

1. to always ensure that $D \in \bar{S}^3_{++}$
2. to reduce the dimensionality of the problem by recasting the original problem (4.17) into a 2D-problem.

The exact methods to achieve both modifications will thoroughly discussed in the next section.

4.6.1 Representation of the conductivity tensor

In principle, after accounting for symmetry, $D(\mathbf{x})$ has 6 component to be identified for every $\mathbf{x} \in \Omega$. Since the dynamic of the wave propagation is bound to the 2-D manifold, however, the component normal to the surface does not influence the solution. We therefore define $D(\mathbf{x})$ as follows:

$$D(\mathbf{x}) = P(\mathbf{x}) \begin{pmatrix} \tilde{D}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0}^\intercal & 0 \end{pmatrix} P(\mathbf{x})^\intercal, \tag{4.19}$$

where $\tilde{D}(\mathbf{x}) \in S^2_{++}$ and $P(\mathbf{x})$ is a rotation from the canonical base in $\mathbb{R}^3$ to a local base $\{\mathbf{v}_1(\mathbf{x}), \mathbf{v}_2(\mathbf{x}), \mathbf{n}(\mathbf{x})\}$. The local base at $\mathbf{x} \in \Omega$ is such that $\mathbf{n}(\mathbf{x})$ is the normal vector to the surface and the orthogonal vectors $\mathbf{v}_1(\mathbf{x})$, $\mathbf{v}_2(\mathbf{x})$ span the tangent space. In such way, the dimension of the parameter space is reduced from 6 to only 3. Any basis $\mathbf{v}_1(\mathbf{x})$ in the tangent space is valid, but we want to have a smooth basis that minimizes the variation across the manifold:

$$\min_{\mathbf{v}_1} \int_{\Omega} ||\nabla_{\mathcal{M}} \mathbf{v}_1(\mathbf{x})||_2^2 \, \mathrm{d}\mathbf{x} \;\text{ s.t. }\; ||\mathbf{v}_1(\mathbf{x})|| = 1, \tag{4.20}$$

In PIEMAP, Section 4.6.2, we minimized (4.20) directly, whereas for Section 4.6.3, we used the vector heat method [113] that computes the parallel transport of a single given vector. The need for this smooth basis is explained by the use of TV regularization on the parameters that define the conduction velocity on the manifold:

$$\mathrm{TV}_{\varepsilon,\lambda}(\cdot)\lambda \int_{\Omega} H_{\varepsilon}(\nabla_{\mathcal{M}} \cdot) \, \mathrm{d}\mathbf{x} \tag{4.21}$$

Where for H_α, we recall the Huber function given by:

$$H_\alpha(\mathbf{x}) = \begin{cases} \frac{1}{2\alpha} ||\mathbf{x}||^2, & \text{if } ||\mathbf{x}|| < \alpha, \\ ||\mathbf{x}|| - \frac{1}{2}\alpha, & \text{else} \end{cases} \tag*{((2.64) revisited)}$$

As already mentionted, the conductivity tensor D was defined through a parameter vector in a way that ensures its symmetry, positive-definiteness, and zero velocity orthogonal to the atrial surface $\mathcal{M} \subset \mathbb{R}^3$ in all cases (see (4.18)). We therefore consider the parameter vector $\mathbf{d}(\mathbf{x}) = [d_1(\mathbf{x}), d_2(\mathbf{x}), d_3(\mathbf{x})]^\top$ and define D through $\mathbf{d}$ as follows:

$$D(\mathbf{x}) := e^{P(\mathbf{x}) D_2(\mathbf{d}(\mathbf{x})) P(\mathbf{x})^\top}, \qquad D_2(\mathbf{d}(\mathbf{x})) := \begin{bmatrix} d_1(\mathbf{x}) & d_2(\mathbf{x}) \\ d_2(\mathbf{x}) & d_3(\mathbf{x}) \end{bmatrix} \tag{4.22}$$

where $P(\mathbf{x}) \in \mathbb{R}^{3\times 2}$ is a matrix whose columns contain the two orthonormal vectors $\mathbf{v}_1$ $\mathbf{v}_2$ in the tangent plane at $\mathbf{x} \in \mathcal{M}$. This choice corresponds to the Log-Euclidean metric of the space S^3_{++} [5], where the matrix exponential is computed from the eigendecomposition of D_2. In particular, the admissible parameter set $\mathbf{d} \in \mathbb{R}^3$ is mapped through (4.22) to $\bar{S}^3_{++}$. This space offers beneficial properties for interpolating tensors and also showed quick convergence results in our experiments. The main benefit however is the fact that without any projection, our function always fulfills $D(\mathbf{d}) \in \bar{S}^n_{++}$. For more detailed explanations about the Log-Euclidean space, we refer to [5].

The fiber direction is defined as the direction of fastest propagation, that is the eigenvector associated to the largest eigenvalue of D. By the definition of matrix exponential, this direction is also the maximum eigenvector of $PD_2P^\top$.

4.6.2 PIEMAP: Personalized Inverse Eikonal Model from cardiac Electro-Anatomical Maps

Here we briefly summarize our method presented in [61]. PIEMAP also computes the gradient by solving the eikonal equation using the FIM and deriving its result. Unlike

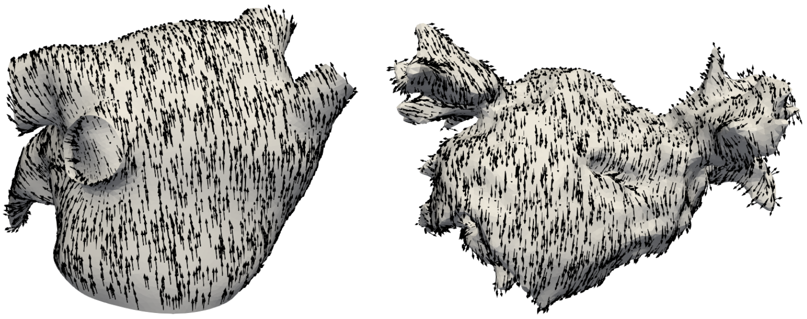


Figure 4.12: Smooth generated manifold bases using the vector heat method for both considered models. These bases are used to provide a smooth 2D map across the manifold and are a useful foundation for computing TV in 2D.

Section 4.5 however, we use here a simplified version of the FIM and employ automatic differentation.

4.6.2.1 Methods

Forward Problem As already mentioned, in [52] an efficient method (FIM) is given to solve (4.17). However, to exploit common ML frameworks and to ease implementation, we consider that the Active List in Algorithm 4.1 always encompasses all vertices. Additionally, we exchange the minima functions in (4.3) and (4.7) with softminima functions smin, commonly found in many ML applications to make the function continuously differentiable:

$$\mathrm{smin}^{\kappa}(\mathbf{x}) = -\frac{1}{\kappa} \log \left(\sum_i \exp(-\kappa x_i) \right).$$

The resulting fixed-point iteration scheme $\phi^{k+1} = F(\phi^k)$ can then be shortly and concisely summarized by the update step:

$$\phi_i^{k+1} = \begin{cases} 0, & \text{if } \mathbf{x}_i \in \Gamma_0, \\ \mathrm{smin}^{\kappa}_{T_j \in \omega_i} \mathrm{smin}^{\kappa}_{\mathbf{y} \in e_{i,j}} \left\{ \phi(\mathbf{y}) + \sqrt{\left\langle D_j^{-1}(\mathbf{y} - \mathbf{x}_i), \mathbf{y} - \mathbf{x}_i \right\rangle} \right\}, & \text{otherwise.} \end{cases} \tag{4.23}$$

where ω_i is the patch of triangles T_j connected to the vertex $\mathbf{x}_i$, $e_{i,j}$ is the edge of the triangle T_j opposite to the vertex $\mathbf{x}_i$, $D_j = D|_{T_j}$. This means, differently from the classic FIM method [52], we concurrently update *all* the nodes, that is the map F is applied in parallel to each node and not just on a small portion of "active" nodes. The switch to the soft-minimum smin^{κ} also ensures a limited degree of smoothness of the function and avoids discontinuities in the gradient computation.

PIEMAP and inverse problem PIEMAP implements an inverse problem in which the optimal conductivity tensor field D for (4.17) is selected such that the mismatch between recorded activation times $\hat{\phi}(\mathbf{x})$ and the simulated activation times $\mathrm{FIM}_D(\mathbf{x})$ on the measurement domain $\Gamma \subset \mathcal{M}$ is minimized in the least-squares sense. The inverse problem, therefore, consists in finding the vector field $\mathbf{d} \in \mathbb{R}^3$, which minimizes the following objective function:

$$\min_{\mathbf{d}} \underbrace{\frac{1}{2} \int_{\Gamma} \left(\mathrm{FIM}_{D(\mathbf{d})}(\mathbf{x}) - \hat{u}(\mathbf{x})\right)^2 \mathrm{d}\mathbf{x}}_{U(\mathbf{d})} + \underbrace{\lambda \int_{\Omega} H_{\varepsilon}(\nabla_{\mathcal{M}} \mathbf{d})\, \mathrm{d}\mathbf{x}}_{\mathrm{TV}_{\varepsilon,\lambda}(\mathbf{d})}, \tag{4.24}$$

where $\mathrm{TV}_{\varepsilon,\lambda}(\mathbf{d})$ is the already mentioned smooth total variation (TV) regularization term (Section 4.6) which alleviates the ill-posedness of the problem. The computed 2D-basis of the manifold, computed by minimizing (4.20), is shown in Figure 4.13. We use these bases for all experiments in this section. The need and background for the local bases was more closely discussed in Section 4.6.1. The optimal choice of regularization parameter λ is obtained by using a cross-validation approach.

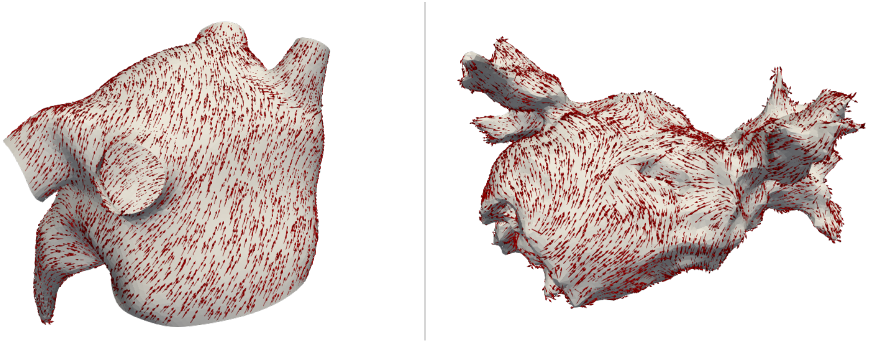


Figure 4.13: Generated local bases on the atria manifold models.

Forward-Backward Splitting and Numerical Solution The computational complexity of solving Equation (4.24) is dominated by the time for computing FIM_D and $\nabla_{\mathbf{d}}\mathrm{FIM}_D$. The implementation of FIM_D in TensorFlow allows for an efficient computation of $\nabla_{\mathbf{d}}\mathrm{FIM}_D$ via backpropagation on a GPU. While the minimization of the residual $U(\mathbf{d})$ is usually achieved very quickly, at least when $\hat{\phi}$ is a (possibly corrupted) solution of (4.17), the TV term tends to increase the number of needed iterations for convergence. In order to increase the convergence rate, we apply the principle of the FISTA [12], quadratically bounding the non-linear, non-convex function U around the current point $\mathbf{d}_k$ by appling the descent lemma from Theorem 2.3:

$$U(\mathbf{d}) \leq U(\mathbf{d}_k) + \langle \nabla_{\mathbf{d}} U(\mathbf{d}_k), (\mathbf{d} - \mathbf{d}_k) \rangle + \frac{L}{2} \left|\left|\mathbf{d} - \mathbf{d}_k\right|\right|_2^2 =: G(\mathbf{d}). \tag{4.25}$$

Recall that the bounding function $G(\mathbf{d})$ is convex, hence has a unique minimum $\bar{\mathbf{d}} = \mathbf{d}_k - L^{-1}\nabla_{\mathbf{d}}U(\mathbf{d}_k)$, as was already discussed in Section 2.10 and Definition 2.21 in particular. As $\mathrm{TV}_{\varepsilon,\lambda}(\mathbf{d})$ is also convex, we obtain the following convex minimization problem:

$$\min_{\mathbf{d}} \frac{L}{2}\|\mathbf{d} - \bar{\mathbf{d}}\|_2^2 + \mathrm{TV}_{\varepsilon,\lambda}(\mathbf{d}).$$

We recast the problem into a convex-concave saddle-point problem:

$$\min_{\mathbf{d}} \max_{\mathbf{p}} \frac{L}{2}\left|\left|\mathbf{d} - \bar{\mathbf{d}}\right|\right|_2^2 + \langle \nabla_{\mathcal{M}}\mathbf{d}, \mathbf{p}\rangle - \mathrm{TV}^*_{\varepsilon,\lambda}(\mathbf{p}) \tag{4.26}$$

where $\mathrm{TV}^*_{\varepsilon,\lambda}$ denotes the convex conjugate of our smoothed TV. The problem can now be solved using the Primal-Dual algorithm [25] given by:

$$\begin{cases} \mathbf{d}^{i+1} = \mathrm{prox}_{\tau G}(\mathbf{d}^i - \tau\nabla^*_{\mathcal{M}}\mathbf{p}^i) \\ \mathbf{d}_\Theta = \mathbf{d}^{i+1} + \theta\left(\mathbf{d}^{i+1} - \mathbf{d}^i\right) \\ \mathbf{p}^{i+1} = \mathrm{prox}_{\sigma \mathrm{TV}^*_{\varepsilon,\lambda}}\left(\mathbf{p}^i + \sigma\nabla_{\mathcal{M}}\mathbf{d}_\Theta\right) \end{cases} \tag{(2.70) revisited}$$

with

$$\hat{\mathbf{d}} = \underset{\tau U}{\mathrm{prox}}(\tilde{\mathbf{d}}) = \left(\tilde{\mathbf{d}} + \tau L\bar{\mathbf{d}}\right) / (\tau L + 1)$$

$$\hat{\mathbf{p}} = \underset{\sigma \mathrm{TV}^*_{\varepsilon,\lambda}}{\mathrm{prox}}(\mathbf{p}) \Leftrightarrow \hat{\mathbf{p}}_j = \begin{cases} \frac{\bar{\mathbf{p}}_j}{||\bar{\mathbf{p}}_j||/\lambda} & \text{if } ||\bar{\mathbf{p}}_j|| > 1 \\ \bar{\mathbf{p}}_j & \text{else} \end{cases}$$

for $\bar{\mathbf{p}}_j = \frac{\mathbf{p}_j}{\sigma\varepsilon/\lambda+1}$, $\theta = 1$ and $\tau\sigma\,||\nabla_{\mathcal{M}}||_2^2 \leq 1$. The parameter L in Equation (4.25), usually challenging to evaluate, is computed through a Lipschitz backtracking algorithm [12].

4.6.2.2 Experiments

For the evaluation of PIEMAP, we first assessed its effectiveness on reconstructing known conduction velocity and fibers on a realistic human LA model, also in the presence of white noise and heterogeneity. The LA model was generated from MRI data of a patient, with the fibers semi-automatically assigned as described previously [55]. Fiber and transverse velocity were set to 0.6 $\frac{\mathrm{m}}{\mathrm{s}}$and 0.4 $\frac{\mathrm{m}}{\mathrm{s}}$respectively for the entire LA, except for the low conducting region, where we used 0.2 $\frac{\mathrm{m}}{\mathrm{s}}$for both fiber and transverse velocity. We tested PIEMAP both in the case of fully anisotropic and in the case of isotropic conduction. In the latter case, in particular, we compared PIEMAP to existing methods for the evaluation of conduction velocity, namely a local method [23] and EikonalNet [109], a PINN method. In a second set of experiments, we eventually applied PIEMAP to clinically acquired data, in the form of high-density EAM.

All examples were optimized for 2000 iterations, with each iteration taking about 1.8

seconds, totalling into a run-time of approximately 1 hour for one optimization.

Numerical assessment All the experiments were performed on a human, cardiac magnetic resonance derived LA model, with semi-automatically placed fiber directions based on histological studies. The Ground-Truth (GT) solution was computed with a single earliest activation site using (4.17), and with a low-conducting area being close to the left atrial appendage. Different levels of independent and identically distributed (i.i.d.) Gaussian noise with standard deviation $\sigma_{\mathcal{N}}$ were tested. The measurement domain was a set of 884 points uniformly distributed across the atrium. All models are compared to a local approach that computes front velocity and direction from a triangulation of the measurement point cloud. We compare the local approach with our method for the 3D in-silico LA model. The reconstruction root-mean-square error (RMSE) with respect to GT was evaluated in terms of conduction velocity (m/s), propagation direction and, only for PIEMAP, fiber-angle error. To evaluate the results, we compute the front direction and fiber direction unit vectors, denoted as $\mathbf{e}$ and $\mathbf{f}$ respectively. The front and fiber angle-errors are then defined as $\alpha_{\mathbf{e}} = \arccos \langle \mathbf{e}, \mathbf{e}_{\mathrm{GT}} \rangle \in [0, 180°)$ and $\alpha_{\mathbf{f}} \arccos |\langle \mathbf{f}, \mathbf{f}_{\mathrm{GT}} \rangle| \in [0, 90°)$. The velocity errors in propagation direction are then $v \cos(\alpha_{\mathbf{x}}) - v_{\mathrm{GT}}$ for computed velocity v and exact velocity v_{GT}, both in the front and fiber direction.

Table 4.2: Comparison of the front-velocity/front-angle error of PIEMAP with the local approach [23] and EikonalNet [109], assuming different noise levels for the in-silico models. Errors in $\frac{m}{s}$/degree. In the last column, we compare the fiber velocity error and fiber angle error.

		Error in Propagation Direction			**Fiber Error**
		PIEMAP	Local Method	EikonalNet	PIEMAP
$\sigma_{\mathcal{N}}$/PSNR	0ms/∞ dB	**0.20**/10.58	**0.20**/22.95	0.53/**9.20**	0.25/38.34
	0.1ms/64.1 dB	**0.19**/10.61	0.20/23.17	0.40/**9.10**	0.25/38.46
	1ms/43.9 dB	**0.20/11.03**	0.21/23.57	0.49/14.60	0.25/38.59
	5ms/29.9 dB	**0.25/19.94**	0.29/30.20	1.24/49.40	0.26/40.14

Results are reported in Table 4.2. All methods correctly captured the low conduction region. PIEMAP compared favourably to the local method at all noise levels in terms of absolute conduction velocity. EikonalNet shows a slightly more accurate front angle error, which is counteracted by the considerably high front velocity error, both compared to our and the local method.

Overall, PIEMAP had the benefit over EikonalNet that the GT was generated with the anisotropic eikonal model, and thus it is in theory possible to reproduce the data exactly with a zero noise level. In the local method no model assumption is made. Interestingly, the error in front direction for the local method could be linked to the fact that, in the presence of anisotropic conduction, propagation direction and $\nabla_{\mathcal{M}}\phi$ differ. For instance, a circular propagation from the source x_0 satisfies (2.23) with $\phi(x) =$

$\sqrt{\langle D^{-1}(x-x_0),(x-x_0)\rangle}$, thus $\nabla_{\mathcal{M}}\phi$ differs from $x-x_0$, which is the propagation direction. In the local method, $\nabla_{\mathcal{M}}\phi$ is used to establish such direction. In EikonalNet, results were less robust to noise. A plausible explanation is that training Neural Networks does not always yield the same results, as multiple local minima might be present. Therefore, error can be slightly lower or higher depending on the initial conditions. In terms of computational time, PIEMAP was comparable to EikonalNet, but significantly slower than the local method.

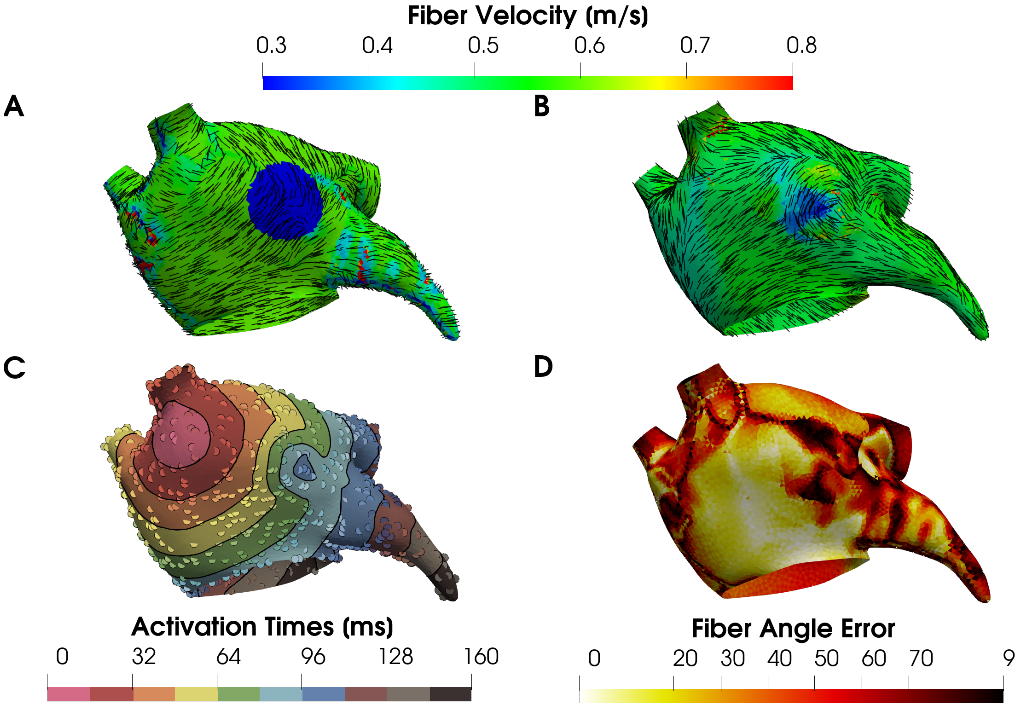


Figure 4.14: Results of our method on the noise-less LA-model with known ground-truth fiber orientation and velocity (A). The scarred region is correctly activated at a later time by a combination of reducing fiber-velocity, as well as aligning fibers along the contour lines (B/C). The activation map can faithfully capture the observed measurement points $u(\mathbf{x} \in \Gamma)$, marked as colored dots (C) and matches the GT-model's activation closely (not shown). Fiber alignment of our model (B) shows mostly errors around the mitrial valves, the scarred region and pulmonary veins (D), as well as regions of high curvature. Best viewed online.

Regarding the reconstruction of fiber directions (see Figure 4.14), we observed a very good performance for the fiber and cross-fiber velocity, and a reasonable reconstruction for the direction. In particular, reconstruction in fiber direction was poor around the boundaries (mitral ring and pulmonary veins, where fibers run parallel to the opening) and in the scarred region, which attribute the most to the fiber angle error in Table 4.2. The distribution of fiber angle errors is a slightly left-skewed uniform distribution (not shown), indicating that the chosen smooth basis along with a simple TV prior can provide resonable results with respect to the activation timings. Still, it may not be sufficient to account for the partly complicated fiber orientation, especially in areas of high-curvature of the mesh or sudden changes of fiber orientation on the endocardium as an effect of the volumetric structure of the atria, such as is the case for the mitrial valve. Physiological priors will need to be considered in the future for this purpose.

Application to real clinical data In a patient candidate to ablation therapy, a high-density activation map along with a 3D patient-specific atrial model was acquired with an EAM system (Catheter: Pentaray® System: CARTO® 3 System, Biosense Webster). The recordings encompassed roughly 850 "beats" of 2.5 sec including both the electrode position in 3D space and the unipolar electrogram (1 kHz). Recordings that were deemed to be untrustworthy due to 1) insufficient contact, 2) sliding of the electrode in 3D space >1 cm, 3) correspondence to a inconsistent surface P-wave, 4) minimal unipolar amplitude, were excluded automatically from the study. To avoid degenerated triangles with acute angles, sometimes created by the EAM recordings, we used PyMesh[1] to postprocess the mesh. A further manual pre-processing of the signals was eventually performed for a correct detection of the local activation time (steepest negative deflection in the unipolar signal) in the last beat and compared to local bipolar signals for confirmation. Distribution of points was uneven across the LA, as many points were located around the pulmonary veins.

Of the remaining valid 565 beats, randomly chosen 80% (452 points) were used to optimize Equation (4.24), while the remaining 20% were used as a cross-validation set to find the optimal regularization parameter λ. Figure 4.15 shows the fiber velocity, orientation and activation map after the optimization. The cross-validation error over several values of λ is shown in Figure 4.16, which lead us to the used value of λ. The best cross-validation error lead to a relatively smooth fiber velocity field, with velocities ranging up to $1.5\frac{\mathrm{m}}{\mathrm{s}}$ in the initiation region, probably a consequence of choosing only one mesh node as an initiation site when in reality the initiation site is larger or composed of multiple sites. A speed-up of propagation near the atrial wall can often be witnessed and is compensated in our model by an overall higher fiber-velocity.

4.6.2.3 Extended Evaluation on Clinical Data

We additionally tested our approach PIEMAP on more clinically acquired data in [83]. More specifically, we tested on 9 different patient EAMs acquired during clinical intervention. Information on the statistics of the recordings can be found in Table 4.3.

The numerical results of of the patient recordings are presented in Table 4.4. With the exception of patients 4 and 6, we received a very high correlation with the measurement points that we optimized for. Similarly to the experiments in Section 4.6.2.2, we used a randomly drawn validation set of 20% that we also tested the correlation on (rightmost

[1] `https://github.com/PyMesh/PyMesh`

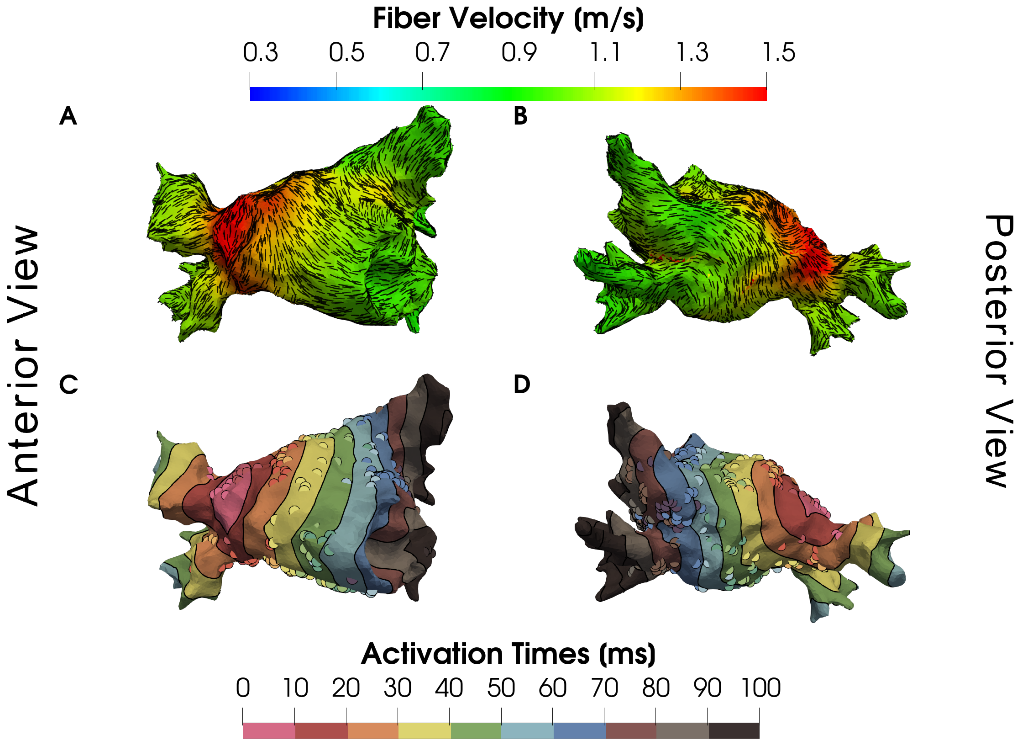


Figure 4.15: Anterior (A/C) and posterior view (B/D) of PIEMAP's results on a patient's left atria. The panels A and B show the found fiber direction and fiber velocity, while the panels C and D show the activation map along with the actual measured points on top, similar to Figure 4.14.

column) and gained similarly high correlation. We additionally validated PIEMAP on small spatial regions (clusters) that were left out from the optimization. We distinguished between early, middle and late clusters according to their LAT. The results for these validations are also visible in Table 4.4: Discarding early clusters tends to have a very negative impact on the optimization and subsequently the correlation at these regions, while late clusters showed a high correlation for all but the two cases with low coverage.

Figure 4.17 shows a qualitative evaluation of PIEMAP, with the reconstructed LAT map, conduction velocities and fiber orientation estimation for patients 3, 5, and 7. The computed optimal conduction velocity distribution is shown on the bottom and is centered around 0.6m/s. More information on all details of the evaluations can be found in the original paper [83].

4.6.3 Anisotropic Eikonal Net

We already extensively discussed in Section 2.9 how commonly encountered PDEs can be solved using the FEM, as was done in Section 4.6.2. A very recent alternative approach from [100] involves the shortly mentioned PINN, combining a PDE-constrained optimiza-

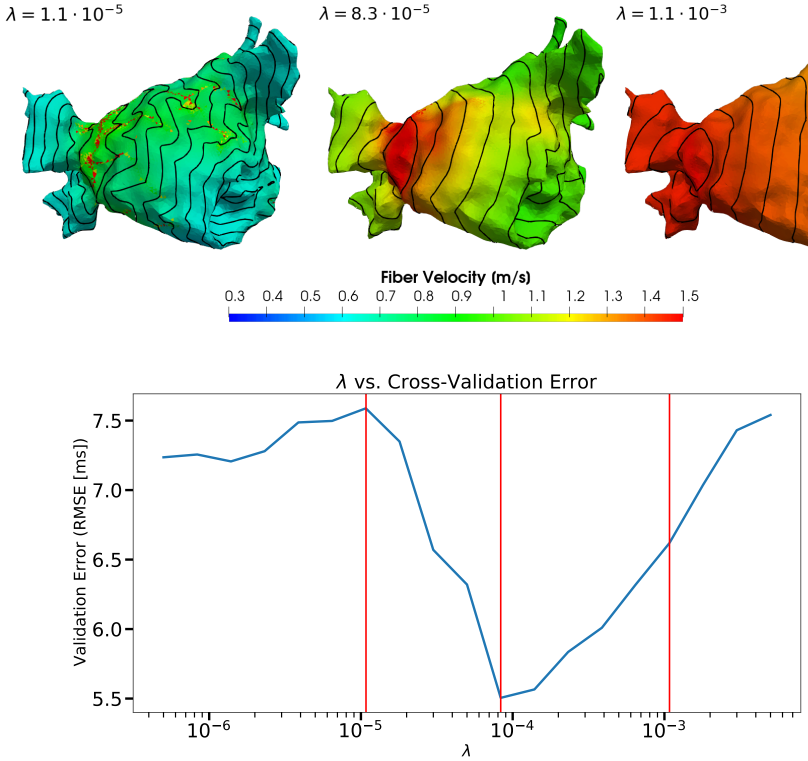


Figure 4.16: Influence on the final cross-validation error when varying over λ for the optimization on the EAM recordings. The black contours are the isochrones of the modelled activation. Shown are the results of three different λ values: The left result is the least regularized with $\lambda = 1.1 \cdot 10^{-5}$, while the result on the right side is heavily regularized with $\lambda = 1.1 \cdot 10^{-3}$. A compromise is the figure in the middle with $\lambda = 8.3 \cdot 10^{-5}$, but finding the physiologically most plausible value for λ is a non-trivial task since we do not know the true distribution of velocities and activations in the atria.

tion with neural-network like structures. Consider a fully connected neural network in with the cartesian coordinates as inputs (in the case of EAMs 3D, x, y, z), used to calculate the value of the PDE function $\phi : \Omega \rightarrow \mathbb{R}$. The resulting neural network thus tries to approximate the function $\phi_{\text{NN}} : \mathbb{R}^3 \rightarrow \mathbb{R}$. The classical ML paradigm would suggest to learn this function directly from data using a data fidelity term $\mathcal{D}$ and a regularizing term $\mathcal{R}$, but typically fails to do so in the absence of a massive amount of data. In contrast, in PINNs, we compute the gradient $\nabla\phi$ by using standard means of backpropagation, used in computing gradients of ANNs, and use $\nabla\phi$ in an additional loss term, mimicking the desired PDE. This special loss term is in practice evaluated at a discrete set of collocation points. The benefits of such a method were already extensively discussed and demonstrated in [100]: The amount of data needed to effectively optimize a PDE function with high-fidelity is massively reduced since collocation points can be chosen at will on the domain Ω and theoretically without any need of data. While the FEM heavily depends both

Patient	# Beats	# Rec. (rec./beat)	Valid rec.	Coverage (%)
1	276	699 (2.5)	517 (74%)	71
2	458	1211 (2.6)	993 (82%)	73
3	392	856 (2.2)	685 (80%)	80
4	419	1104 (2.6)	905 (82%)	35
5	638	1657 (2.6)	1309 (79%)	77
6	199	414 (2.1)	319 (77%)	61
7	189	421 (2.2)	261 (62%)	50
8	265	822 (3.1)	612 (75%)	80
9	646	1829 (2.8)	1500 (82%)	82
Overall	387 ± 173	1001 ± 499	789 ± 425	68 ± 16

Table 4.3: EAM characteristics of the 9 patients that PIEMAP was tested on. The final valid recordings were manually acquired by discarding unreliable measurements. Overall coverage was good with the exception of patients 4 and 7. Table from [83].

Patient	Reconstruction error				Validation			
	PIEMAP vs. measurements		PIEMAP vs. interpolation		Cluster			Sparse
	Corr	Error (mV)	Corr	Error (mV)	Early	Mid	Late	
1	0.93	5.30	0.86	8.83	−0.46	0.50	0.56	0.87
2	0.96	5.52	−0.21	528.28	−0.25	0.69	0.58	0.95
3	0.97	4.58	0.89	10.03	0.84	0.92	0.80	0.96
4	0.62	22.84	0.09	91.68	−0.07	0.30	−0.29	0.57
5	0.93	6.68	0.92	10.98	0.75	0.69	0.61	0.91
6	0.74	11.84	0.80	20.05	0.57	0.44	0.49	0.66
7	0.96	5.69	0.51	23.43	−0.41	0.13	−0.09	0.88
8	0.97	3.29	0.86	9.82	0.89	0.91	0.69	0.96
9	0.97	4.28	−0.32	1010.49	0.79	0.83	0.78	0.96
Overall	0.96	7.8 ± 6.6	−0.01	190 ± 350	0.77	0.86	0.88	0.95

Table 4.4: Results of applying PIEMAP to the 9 patient cases. Table from [83]. Note that error is actually in [ms].

on the choice of a mesh and the polynomial basis for their solution, PINNs and the order of their solution solely depends on the neural network structure. Note however that the choice of the loss functional for PDEs of order n necessitates that the activation functions are at least in H^{n+1}, i.e. $n + 1$ times weakly differentiable, since the optimization can only be efficiently implemented using first order gradient based methods. Boundary conditions can be weakly imposed and additional inverse loss terms allow to effectively build an inverse PDE problem solver, demonstrated in this section, based on our paper [60].

4.6.3.1 Methods

Anisotropic eikonal model for cardiac activation The eikonal equation is not explicitly solved, hence there is no need to enforce boundary conditions. We rather consider

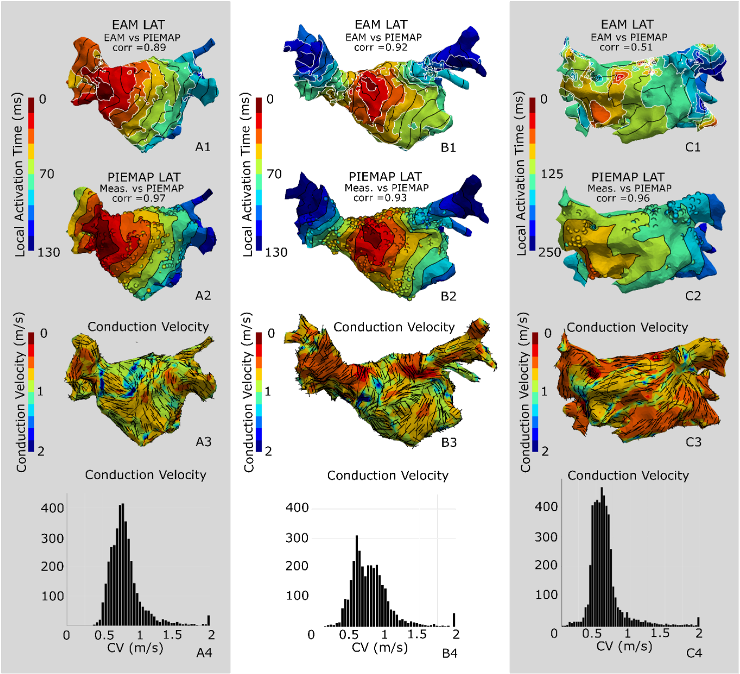


Figure 4.17: Reconstructed LAT maps (second row), conduction velocities and fiber orientations (third row) using PIEMAP on the data of patients 3, 5, and 7 (from left to right). The top row shows the compliance with the LAT map from the Carto software. Figure from [83].

the model residual:

$$R_{\mathrm{m}}[\phi](\mathbf{x}) := \sqrt{\max\left\{D(\mathbf{x})\nabla\phi(\mathbf{x}) \cdot \nabla\phi(\mathbf{x}), \varepsilon\right\}} - 1, \tag{4.27}$$

for a sufficiently small $\varepsilon > 0$ to avoid infeasible gradients, as a metric of point-wise model discrepancy for a given pair of activation ϕ and conductivity tensor D.

Physics-informed neural network In the considered experiments, we are given a set of points, each composed by a location $\mathbf{x}_i$ and a recorded activation time $\hat{\phi}_i : \Gamma_{\mathcal{M}} \to \mathbb{R}$ for $\Gamma_{\mathcal{M}} \subset \mathcal{M}$ representing the EAM locations and timings of the recordings. The objective is therefore to identify a conductivity tensor field D such that the corresponding activation map ϕ, as resulting from (4.17), will closely reproduce the observed data. The tensor D can then be reconstructed by means of using $\mathbf{d}$.

For this purpose, we approximate both the activation map $\phi(\mathbf{x})$ and the conductivity vector $\mathbf{d}(\mathbf{x})$ with a feed-forward neural network $\mathrm{NN}_{n,m,\boldsymbol{\theta}} \colon \mathbb{R}^n \to \mathbb{R}^m$ with n inputs to m outputs and characterized by a vector $\boldsymbol{\theta}$ containing weights and biases, as was initially promoted in [100]. The used architecture of the networks is shown in Figure 4.18. Specifically, we have $\phi(\mathbf{x}) \approx \phi_{\mathrm{NN}}(\mathbf{x}, \boldsymbol{\theta}_\phi) = \mathrm{NN}_{3,1,\boldsymbol{\theta}_\phi}(\mathbf{x})$ and $\mathbf{d}(\mathbf{x}) \approx \mathbf{d}_{\mathrm{NN}}(\mathbf{x}, \boldsymbol{\theta}_{\mathbf{d}}) =$

$d_{\max} \cdot \tanh\bigl(\mathrm{NN}_{3,3,\boldsymbol{\theta}_{\mathbf{d}}}(\mathbf{x})\bigr)$, where tanh is meant component-wise, and $d_{\max}$ is an upper limit for the components of $\mathbf{d}_{\mathrm{NN}}$, meant to avoid over- and underflows in the numerical calculations. This construction of ϕ and $\mathbf{d}$ enables us to use standard ML methods and frameworks to efficiently calculate the gradients $\nabla\phi$ and $\nabla\mathbf{d}$, used in the chosen PDE model (4.27) and inverse regularization. The usage of these first order gradients in the optimization necessitates at least second order weakly differentiable activation functions in the neurons, achieved by the use of tanh functions.

Similar to the original PINN algorithms in [100], we define a loss function to train our model as the sum of a data fidelity, a PDE model fidelity term and two regularization terms:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_\phi, \boldsymbol{\theta}_{\mathbf{d}}) \coloneqq{}& \int_{\Gamma_{\mathcal{M}}} \bigl(\phi_{\mathrm{NN}}(\mathbf{x}) - \hat{\phi}(\mathbf{x})\bigr)^2 \,\mathrm{d}\mathbf{x} + \alpha_{\mathrm{m}} \int_{\mathcal{M}} \bigl(R_{\mathrm{m}}[\phi_{\mathrm{NN}}](\mathbf{x})\bigr)^2 \,\mathrm{d}\mathbf{x} \\ &+ \alpha_\theta \bigl(\|\boldsymbol{\theta}_\phi\|^2 + \|\boldsymbol{\theta}_{\mathbf{d}}\|^2\bigr) + \alpha_{\mathbf{d}} \int_{\mathcal{M}} H_\varepsilon\bigl(\nabla \mathbf{d}_{\mathrm{NN}}(\mathbf{x})\bigr)\,\mathrm{d}\mathbf{x},\end{aligned} \tag{4.28}$$

for the three weighting parameters $\alpha_{\mathrm{m}}, \alpha_{\mathbf{d}}, \alpha_\theta$. Regularization is both applied to the weights of the networks as well as on the inverse parameter estimation. The latter regularization term is, similar to PIEMAP in Section 4.6.2, an Huber-type, approximated TV regularization for the conductivity vector parameters $\mathbf{d}$. Recall that by construction, D has zero velocity in directions normal to the manifold (see Sec. 4.6) and thus allows us to neglect the additional normal penalization used in [109].

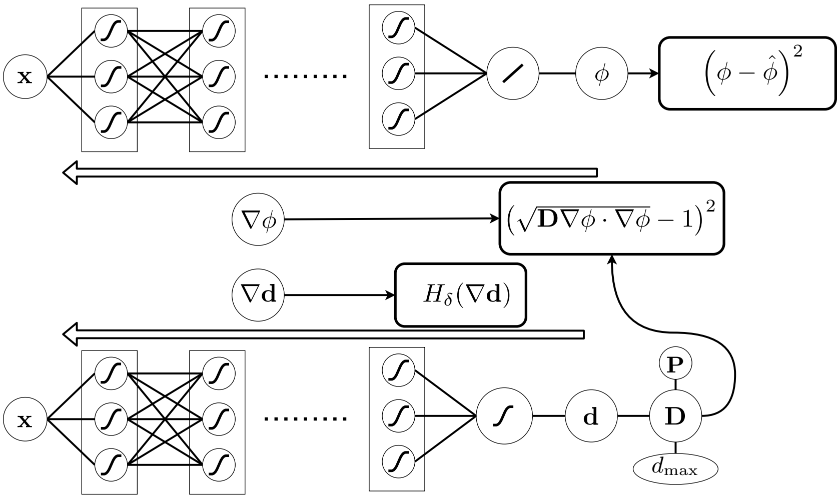

x
ϕ
$(\phi - \hat{\phi})^2$
$\nabla\phi$
$(\sqrt{\mathbf{D}\nabla\phi \cdot \nabla\phi} - 1)^2$
$\nabla\mathbf{d}$
$H_\delta(\nabla\mathbf{d})$
x
d
P
D
$d_{\max}$

Figure 4.18: Structural view of the proposed PINN architecture, containing the two NNs ϕ_{NN} and $\mathbf{d}_{\mathrm{NN}}$. Nodes containing a curve indicate a tanh activation function. The final layer of ϕ_{NN} is a linear layer. D is computed using (4.22). $\nabla\phi$ and $\nabla\mathbf{d}$ can be obtained by means of backpropagation (reverse arrows). The bold rectangular boxes show the three major loss terms from (4.28): Data fidelity, eikonal (PDE) and TV loss (top to bottom).

Numerical implementation The domain $\mathcal{M}$ is discretized using a triangular mesh, usually obtained directly from the mapping system, along with point-wise evaluations of the activation times, that is $\Gamma_{\mathcal{M}} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. In all experiments, the integrals were approximated using a point-wise evaluation for both domains: On the vertices for the approximation of $\mathcal{M}$ and on the discrete measurements for $\Gamma_{\mathcal{M}}$.

For the optimization, we experimentally selected the hyper-parameters as $\alpha_{\mathrm{m}} = 10^4$ for the model atria and $\alpha_{\mathrm{m}} = 10^3$ for the EAM. The other two hyperparameters are the same for both experiments: $\alpha_\theta = 10^{-4}$, $\alpha_{\mathbf{d}} = 10^{-3}$. The two neural networks for ϕ and $\mathbf{d}$ had 7 and 5 hidden fully connected layers respectively. All hidden layers consisted of 20 neurons for ϕ_{NN} and 5 neurons for $\mathbf{d}_{\mathrm{NN}}$, with the weights being initialized using Xavier initialization. This choice of neural network architecture was inspired by the work in [109]. We opted for adding a regression layer to ϕ_{NN}, since this allows us to model arbitrary ranges of ϕ. Optimization is performed by first using the ADAM [79] optimizer for 10^4 epochs with a learning rate of 10^{-3}, followed by a L-BFGS optimization [21] until convergence to a local minimum is achieved. Each experiment took no longer than 1.5 hours on the machine described in Section 4.4.

4.6.3.2 Numerical experiments

Herein, we consider, similarly to PIEMAP (see Section 4.6.2.2), two experiments: a synthetic example, with ground-truth on a realistic anatomy of the left atrium, and an example with patient-specific geometry and data. The first example is optimized and tested against different levels of i.i.d. normal noise: $\tilde{\phi}(\mathbf{x}) = \hat{\phi}(\mathbf{x}) + \mathcal{N}(0, \sigma_{\mathcal{N}})$. We measure the performance of the synthetic model in terms of the root-mean-square error (RMSE) over the whole surface here denoted as $\mathrm{RMSE}_{\mathcal{S}}$. Errors directly on the measurement points, employed in the optimization, are used to compute RMSE_O. In the patient specific example, we randomly split $\Gamma_{\mathcal{M}}$ into Γ_O, used for optimization/training, and Γ_T, for testing.

The results of our method on the in-silico model atria, and a comparison to PIEMAP [61], are presented in Tab. 4.5. Both methods are comparable in terms of RMSE, with both methods achieving less than 5ms of RMSE for all levels of noise. Additionally, we tested the presented PINN on an EAM, achieving a RMSE of the activation times on the test set of $\mathrm{RMSE}_{\mathrm{T}} \approx 5.59\,\mathrm{ms}$. The RMSE on the measurements used in the optimization was only slightly lower at $\mathrm{RMSE}_O \approx 4.82\,\mathrm{ms}$, indicating that α_{m} was chosen in a proper range to avoid overfitting to the data. In the patient-specific test (not shown in the table), our method was able to outperform PIEMAP, which reported $\mathrm{RMSE}_{\mathrm{T}} \approx 6.89$ms and $\mathrm{RMSE}_O \approx 1.18$ms on test and optimization/training set respectively, showing a slight overfit to the data used in the optimization.

Figure 4.19 shows the qualitative results of using this method on the two chosen models (the model atria in the noise-less case). We can nicely fit the activation encountered at the surface and create an eikonal-like activation. The initiation sites are automatically deduced by the PINN algorithm with only soft eikonal and data constraints. The smooth

Table 4.5: Evaluation of the presented PINN approach compared to PIEMAP [61] for different levels of noise (given in standard deviation and signal-to-noise ratio) on the in-silico model atria. The result of the noiseless scenario (∞ dB) is visualized in Figure 4.19 on the left side.

		$\text{RMSE}_{\mathcal{S}}/\text{RMSE}_{\mathcal{O}}$ PINN	$\text{RMSE}_{\mathcal{S}}/\text{RMSE}_{\mathcal{O}}$ PIEMAP
$\sigma_{\mathcal{N}}$/PSNR	0 ms/∞ dB	2.20/1.38	1.04/0.83
	0.1 ms/64.1 dB	4.28/2.08	1.02/0.83
	1 ms/43.9 dB	3.32/1.39	1.09/0.83
	5 ms/29.9 dB	3.76/1.85	1.90/0.84

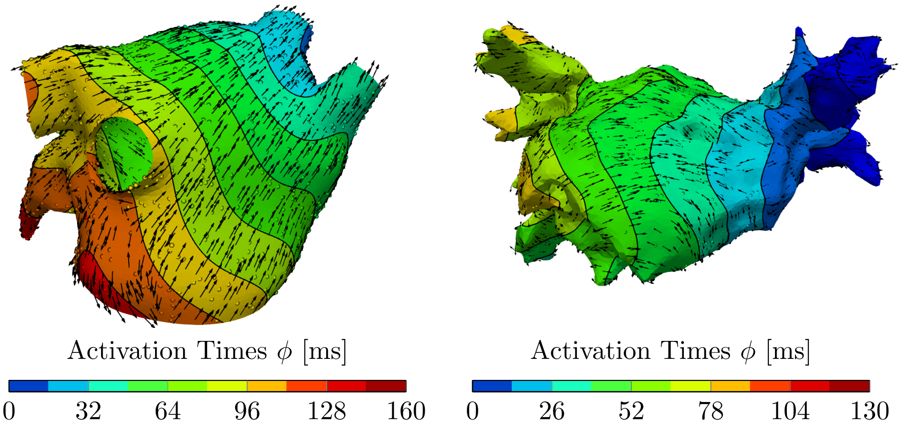


Figure 4.19: Results of the PINN method on an in-silico (left) and an in-vivo (right) EAM model with the overlayed measurements as points and fibers as arrows. The underlying contour lines and colors on the mesh itself represents the activation of the PINN, sampled at each vertex.

basis generated with the vector heat method, together with the TV regularization give us a smooth, fiber field.

4.7 Identifying cardiac initiation sites through geodesics - GEASI

In [58], we address the central question of identifying EASs. In a healthy (human) heart, the spontaneous electric activity of the sino-atrial node initiates an electric wave that first spreads across the atria, and then propagates through the ventricles via the atrio-ventricular node/His bundle/Purkinje network pathway. In patient-specific models, the EASs determine many physiological phenomena such as the Purkinje entry points. For these phenomena, a proper manual quantification is in practice infeasible, especially in the presence of pathological conditions such as bundle branch block, ventricular tachycardia or premature ventricular contractions. Thus, inter-patient variability of electrical activation necessitates automatic individual parameter identification. The identification of earliest activation sites, possibly from non-invasive or minimally-invasive recordings, is therefore of clinical relevance for a correct diagnosis and optimal treatment.

Indeed, a grand challenge in cardiac modeling for personalized health care is the aforementioned individualization of the model parameters for given patient data, which may include standard imaging, ECG, or invasive mapping. A particularly suitable model that exposes EASs is the eikonal equation, which was originally exploited as a convenient approximation of the monodomain and bidomain models [32, 77] and is nowadays more often utilized for its computational efficiency [87]. In this work, we present a novel gradient-based approach for localizing the EASs termed GEASI (**G**eodesic-based **E**arliest **A**ctivation **S**ites **I**dentification). We start from the anisotropic eikonal equation as a common model for cardiac electrophysiology [33], in which the EASs define boundary conditions at specific sites. The anisotropy arises from the fiber alignment inside the heart [32]. The main goal of our approach is the minimization of a given objective functional depending on the solution of the anisotropic eikonal equation as a function of the EASs. Here, a feasible optimization strategy involves the Hamilton–Jacobi formalism, which promotes a tractable derivative with respect to the EASs [37]. Note that this derivative is geometrically related to the tangent of the geodesic at the EASs. In this respect, a geodesic connects an EAS such as a Purkinje entry point to an observation through a path of minimum distance in a predefined metric. Finally, we exploit the aforementioned methods to introduce GEASI, which in its core employs a quadratic mismatch between the eikonal solution and the measurements in the objective functional.

We emphasize that our method is not limited to this quadratic objective functional and can straightforwardly be extended to other scenarios. In this work, we additionally investigate two such extensions: the topological gradient and the fitting of an eikonal solution to a target ECG.

The topological gradient allows for an estimation of the optimal number of EASs. In detail, the concept of topological gradient can be readily introduced via the Hamilton–Jacobi theory. Here, we consider a splitting of a single EAS into a pair of two EASs symmetrically arranged at infinitesimal distance along a given direction in a dipole-like fashion. Thus, the topological gradient provides a criterion to decide if an EAS should be split. In particular, this approach promotes a simple model as a starting point with too little complexity to represent the measurement data and increase the number of source sites until the encountered activations are properly approximated.

In combination with the bidomain equation and the lead field theory, the eikonal model also results in an almost-real-time ECG simulator [95] with remarkable physiological accuracy [33]. Efficient combinations with other methods exist to use solutions of the eikonal equation to compute surface ECGs and BPSMs [95], [87], making these methods a very powerful computational tool. We here employ the existing approach [95] to solve the inverse ECG problem, i.e. we introduce a method to localize EASs purely from ECG data. Numerical experiments in Section 4.7.4 demonstrate that the proposed approach is capable of finding the optimal EASs even in high-fidelity cardiac models.

4.7.1 The GEASI Method

This section introduces the GEASI method, which encompasses the following ingredients. In Section 4.7.1.1, we review the anisotropic eikonal equation and its associated Hamilton–Jacobi formulation. Subsequently, in Section 4.7.1.2 we analyze a general objective function involving the solution of the anisotropic eikonal equation from a functional-analytical perspective. Section 4.7.1.3 deals with the gradient computation of the distance function, which is later exploited in the aforementioned objective functional. Finally, all introduced concepts are combined in Section 4.7.1.4 to define GEASI.

4.7.1.1 Eikonal equation

We again consider the computational domain $\Omega \subset \mathbb{R}^d$ for $d \geq 2$, which in most cases represents the myocardium. Further, let $\mathcal{E}$ be the subset of N pairs $\{(\mathbf{x}_i, t_i)_{i=1}^N\} \in \mathcal{U}_N \coloneqq \Omega^N \times (T_{\min}, T_{\max})^N$ for a priori given $T_{\min} < T_{\max}$ and fixed N. Throughout this work, $\mathcal{E}$ is a set of EASs, where N is the number of EASs, $\mathbf{x}_i$ and t_i are the location and timing of the i-th site, respectively. Let $\phi_{\mathcal{E}} : \Omega \to \mathbb{R}$ be the unique solution of the anisotropic eikonal equation with prescribed values on $\mathcal{E}$, which is commonly referred to as the activation map. Hence, $\phi_{\mathcal{E}}$ defines a slight modification of our original anisotropic eikonal equation in (2.23):

$$\begin{cases} \sqrt{D(\mathbf{x})\nabla\phi_{\mathcal{E}}(\mathbf{x}) \cdot \nabla\phi_{\mathcal{E}}(\mathbf{x})} &= 1, \quad \mathbf{x} \in \Omega \setminus \{\mathbf{x}_1, \dots, \mathbf{x}_N\}, \\ \phi_{\mathcal{E}}(\mathbf{x}_i) &= t_i, \quad (\mathbf{x}_i, t_i) \in \mathcal{E}, \end{cases} \tag{4.29}$$

where we additionally assume now $D \in C^1(\overline{\Omega}, S^d_{++})$. Recall that S^d_{++} is defined as the set of positive definite and symmetric $d \times d$-matrices, which gives rise to the definition of the norm $\|\mathbf{p}\|_D \coloneqq \sqrt{D\mathbf{p} \cdot \mathbf{p}}$ for $\mathbf{p} \in \mathbb{R}^d$. Note that the assumptions already guarantee that

$$\lambda_* \mathbf{I} \prec D(\mathbf{x}) \prec \lambda^* \mathbf{I}$$

for all $\mathbf{x} \in \overline{\Omega}$ and finite bounds $0 < \lambda_* \leq \lambda^* < \infty$. We already extensively discussed in Section 2.8.1 that (4.29) admits a unique viscosity solution according to the theory of Hamilton–Jacobi equations [37]. Here we recall from Section 2.8.1 the definitions geodesics, geodesic distances and length functional associated to the Hamilton–Jacobi. The Lipschitz continuous solution of the eikonal equation $\phi_{\mathcal{E}} \in C^{0,1}(\overline{\Omega})$ is of the form

$$\phi_{\mathcal{E}}(\mathbf{x}) = \min_{(\mathbf{y},t) \in \mathcal{E}} \{t + \delta(\mathbf{x}, \mathbf{y})\} \tag{4.30}$$

where $\delta(\mathbf{x}, \mathbf{y})$ denotes the *geodesic distance*

$$\delta(\mathbf{x}, \mathbf{y}) = \inf_{\widehat{\gamma} \in H^1([0,1], \overline{\Omega})} \{L(\widehat{\gamma}) : \widehat{\gamma}(0) = \mathbf{x}, \widehat{\gamma}(1) = \mathbf{y}\} \tag*{((2.30) revisited)}$$

given the length functional

$$L(\gamma) \coloneqq \int_0^1 \|\dot{\gamma}(t)\|_{D^{-1}(\gamma(t))} \, \mathrm{d}t. \tag*{((2.28) revisited)}$$

Thus, the induced Riemannian metric for two tangent vectors $\mathbf{v}, \mathbf{w}$ is

$$\langle \mathbf{v}, \mathbf{w} \rangle_{\gamma(t)} \coloneqq D^{-1} (\gamma(t)) \, \mathbf{v} \cdot \mathbf{w}. \tag{4.31}$$

We note that the infimum γ in (2.30) is actually attained, and by the geodesic equation we can even deduce $\gamma \in C^{0,1}([0,1], \overline{\Omega})$ (see e.g. [16]). Indeed, in the definition (2.30), we first note that $\|\mathbf{p}\|_{D^{-1}(\mathbf{x})} \leq \lambda_*^{-1} \|\mathbf{p}\|_2$ for all $\mathbf{p} \in \mathbb{R}^d$ and $\mathbf{x} \in \Omega$. Then, for any segment $[\mathbf{x}, \mathbf{y}]$ fully contained in $\overline{\Omega}$ we have that $\delta(\mathbf{x}, \mathbf{y}) \leq \lambda_*^{-1} \|\mathbf{x} - \mathbf{y}\|_2$ since the segment is a geodesic path in the Euclidean norm. Figure 4.20 illustrates a single geodesic path in red on a domain with a continuously varying conduction velocity and isotropic conduction.

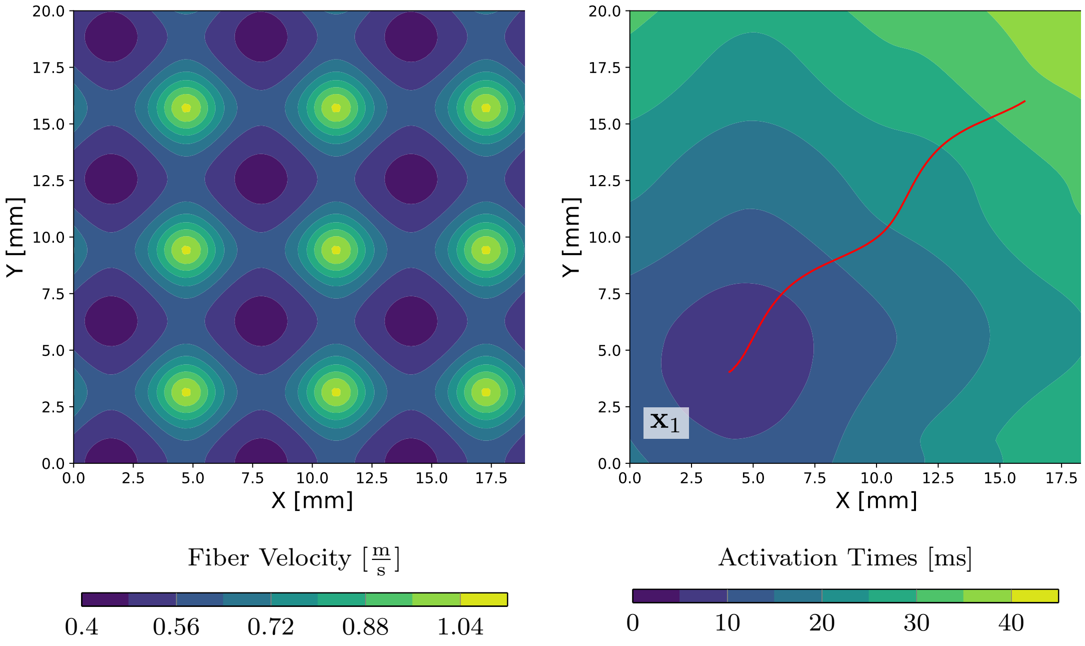


Figure 4.20: Left: fixed velocity field c in Ω. Right: contour plot of the associated anisotropic eikonal equation with anisotropic conduction $D(\mathbf{x}) = c(\mathbf{x})^2 \mathbf{I}$ along with the geodesic path joining the EAS $\mathbf{x}_1$ with an arbitrary point.

When $N > 1$, all pairs $(\mathbf{x}_i, t_i), (\mathbf{x}_j, t_j) \in \mathcal{E}$ must satisfy the subsequent compatibility condition

$$|t_i - t_j| \leq \delta(\mathbf{x}_i, \mathbf{x}_j) \tag{4.32}$$

in order to ensure the existence of a solution. It is worth noting that this condition is not too restrictive. In view of (4.30), which underpins our numerical solution of the eikonal equation, non-compatible data are simply discarded. Interestingly, this is also physiologically sound: a stimulus applied in the depolarized region cannot trigger another propagation.

The Hamilton–Jacobi formulation is essential for computing perturbations of $\mathcal{E}$, which is conducted in the following subsection.

4.7.1.2 Objective functional

The overall objective of GEASI is the minimization of a given functional $\mathcal{J} : C^{0,1}(\overline{\Omega}) \to \mathbb{R}$ depending on the activation map $\phi_{\mathcal{E}}$ with respect to $\mathcal{E}$, i.e.

$$m_N := \min_{\mathcal{E} \in \overline{\mathcal{U}_N}} \mathcal{J}(\phi_{\mathcal{E}}). \tag{4.33}$$

For instance, the objective could describe the minimization of a mismatch (in the least-squares sense) between the simulated activation and the activation detected from epicardial, as well as endocardial mapping (see Section 4.7.4). The objective functional can also involve the activation map implicitly: In Section 4.7.4.3, we utilize the mismatch between the recorded and simulated 12-lead surface ECG as a metric for optimization.

In what follows, we prove the existence of minimizers for (4.33) for varying N. To this end, we define for $\mathcal{E} = \{(\mathbf{x}_i, t_i)_{i=1}^N\}$

$$\Phi_N(\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_N, t_1, \ldots, t_N) := \phi_{\mathcal{E}}(\mathbf{x}). \tag{4.34}$$

Lemma 2. $\Phi_N \in C^{0,1}(\overline{\Omega \times \mathcal{U}_N})$ *is a bounded function of its arguments.*

Proof. Using (4.30), we immediately see that

$$\Phi_N(\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_N, t_1, \ldots, t_N) = \min_{i=1,\ldots,N} \{t_i + \delta(\mathbf{x}_i, \mathbf{x})\}.$$

The Lipschitz continuity of δ as well as the compactness of $\overline{\Omega \times \mathcal{U}_N}$ imply the statement. □

We note that Rademacher's theorem ensures the differentiability of Φ_N almost everywhere. Non-differentiability with respect to $\mathbf{x}$ occurs for instance at $\mathbf{x} = \mathbf{x}_i$, but also in the presence of front collisions. An immediate consequence of this lem is the following

Theorem 4.1 (Existence)**.** *If $\mathcal{J}$ is uniformly continuous, then the problem* (4.33) *admits at least one minimum.*

Proof. The previous lem and the uniform continuity of $\mathcal{J}$ imply the existence of at least one minimum. □

Proposition 1. *Under the hypotheses of Theorem 4.1, m_N is a non-increasing function of N. Moreover, if there exists N such that $m_{N+1} = m_N$, then $m_{N+n} = m_N$ for all $n \geq 1$.*

Proof. The first claim immediately follows from the definition of m_N and set inclusion arguments. To prove the second claim, we assume that $m_{N+1} = m_N$ for some N and $m_{N+2} < m_{N+1}$. However, the choice $\mathbf{x}_{N+1} = \mathbf{x}_{N+2}$ and $t_{N+1} = t_{N+2}$ results in a contradiction. □

Corollary 1. *If N is bounded from above by $N_{\max}$, then $\min_{N \leq N_{\max}} m_N$ has at least one minimum.*

Remark 4.2.

1. *From a practical point of view, this corollary ensures that by adding new EASs, we either improve the objective function or we keep the same level of accuracy. This is also seen in the experiments in Section 4.7.4, where coalescence of two or more sites is observed if introducing too many EASs.*
2. *The minimum in (4.33) is in general not unique as it depends on the choice of $\mathcal{J}$ and on the order of the EASs. In principle, by permuting EASs we obtain the same value of the minimum. In particular, this symmetry induces a periodic partition of the set $\mathcal{U}_N$. Each partition is associated with a specific choice of the order of the EASs. From a numerical point of view, this may constitute a problem for methods based on random sampling. For deterministic steepest descent algorithms, the problem is mitigated by the fact that we rarely cross the boundary between two partitions, e.g., by swapping points, unless the two points coincide.*
3. *In general, we cannot take N unbounded with no further hypotheses on $\mathcal{J}$. Suppose for instance that $\mathcal{J}$ is minimized by $\phi(\mathbf{x}) = c$ for some constant $c \in \mathbb{R}$. Then, $\inf_{N \in \mathbb{N}} m_N$ attains no minimum. Indeed, we cannot represent a constant function with (4.30) if $\mathcal{E}$ is only countable. However, we can approximate the constant with arbitrary precision with a sufficiently large number N of EASs.*

4.7.1.3 Exponential Map

In what follows, we compute the Riemannian exponential map to derive an expression for the variation of the distance function. In particular, we discuss the relation of the derivatives of Φ_N and the geodesic path.

We briefly recall fundamental concepts in Riemannian geometry. Given $\mathbf{x} \in \Omega$ and a tangent vector $\mathbf{v} \in \mathcal{V}$ for a sufficiently small neighborhood $\mathcal{V}$ around the origin of the tangent space at $\mathbf{x}$, the exponential map $\mathrm{Exp}_{\mathbf{x}} : \mathcal{V} \to \Omega$ is given by $\mathrm{Exp}_{\mathbf{x}}(\mathbf{v}) = \gamma(1)$, where $\gamma \in C^{0,1}([0,1], \Omega)$ is a geodesic path with $\mathbf{v} := \dot{\gamma}(0)$. The logarithmic map $\mathrm{Log}_{\mathbf{x}} : \Omega \to \mathcal{V}$ is the inverse of the exponential map $\mathrm{Exp}_{\mathbf{x}}^{-1}$. In other words, the logarithmic map of $\mathbf{y} \in \Omega$

identifies the tangent vector $\dot{\gamma}(0)$ of a geodesic path γ emanating from $\mathbf{x}$ and ending at $\mathbf{y}$.

Proposition 2 (Variation of the distance function)**.** *Let* $\mathbf{x}, \mathbf{y} \in \mathcal{V}$*, where* $\mathcal{V} \subset \mathring{\Omega}$ *is sufficiently small such that all points inside are connected by unique geodesics. Then the variation of* $\delta(\mathbf{x}, \mathbf{y})$ *with respect to* $\mathbf{y}$ *with* $\mathbf{w} = \mathrm{Log}_{\mathbf{x}}(\mathbf{y})$ *reads as*

$$\nabla_{\mathbf{x}} \delta(\mathbf{x}, \mathbf{y}) = -\frac{D^{-1}(\mathbf{y})\mathbf{w}}{\|\mathbf{w}\|_{D^{-1}(\mathbf{y})}}. \tag{4.35}$$

Proof. Suppose that γ is a geodesic with respect to the Riemannian metric in (4.31) realizing the distance $\delta(\mathbf{y}, \mathbf{x})$, i.e. $\gamma(0) = \mathbf{y}$, $\gamma(1) = \mathbf{x}$ and

$$\delta(\mathbf{x}, \mathbf{y}) = \int_0^1 \|\dot{\gamma}(t)\|_{D^{-1}(\gamma(t))} \, \mathrm{d}t.$$

Let $\widetilde{\gamma} : [0, 1] \times (-R, R) \to \mathring{\Omega}$ for small $R > 0$ be a smooth variation of γ such that $\widetilde{\gamma}(t, 0) = \gamma(t)$ for all $t \in [0, 1]$. The first variation formula [91, Chapter 10] with $c = \|\dot{\gamma}(t)\|_{D^{-1}(\gamma(t))}$ for $t \in [0, 1]$ implies

$$\begin{aligned}\nabla_{\mathbf{x}} \delta(\mathbf{x}, \mathbf{y})(\widetilde{\gamma}) = \frac{1}{c}\Bigg(- \int_0^1 \langle \ddot{\gamma}(t), \partial_2 \widetilde{\gamma}(t, 0)\rangle_{\gamma(t)} \, \mathrm{d}t - \sum_{i=1}^{k} \langle \Delta\dot{\gamma}(t_i), \partial_2 \widetilde{\gamma}(t_i, 0)\rangle_{\gamma(t_i)} \\ + \langle \dot{\gamma}(1), \partial_2 \widetilde{\gamma}(1, 0)\rangle_{\gamma(1)} - \langle \dot{\gamma}(0), \partial_2 \widetilde{\gamma}(0, 0)\rangle_{\gamma(0)} \Bigg).\end{aligned} \tag{4.36}$$

Here, $0 < t_1 < \cdots < t_k < 1$ are possible discontinuities of the geodesic curve and $\Delta\dot{\gamma}(t_i) = \dot{\gamma}(t_i^+) - \dot{\gamma}(t_i^-)$, where $\dot{\gamma}(t_i^-)$ and $\dot{\gamma}(t_i^+)$ denote the one-sided derivatives from the left and the right, respectively. The derivative of $\widetilde{\gamma}$ with respect to the second argument is denoted by $\partial_2 \widetilde{\gamma}$. Since γ is assumed to be geodesic and smooth, the first two summands in (4.36) vanish.

By adjusting $\widetilde{\gamma}$ such that $\partial_2 \widetilde{\gamma}(1, 0) = 0$ and observing that $\dot{\gamma}(0) = \mathrm{Log}_{\mathbf{x}}(\mathbf{y})$ we have proven

$$\nabla_{\mathbf{x}} \delta(\mathbf{x}, \mathbf{y})(\widetilde{\gamma}) = -\frac{D^{-1}(\gamma(0))\dot{\gamma}(0)}{\|\dot{\gamma}(0)\|_{D^{-1}(\gamma(0))}} \cdot \partial_2 \widetilde{\gamma}(0, 0),$$

which readily implies (4.35). □

In Proposition 2, we assumed uniqueness and smoothness of the geodesic curve, which is in general not ensured. In practice, the influence of geodesics violating these assumptions is negligible and thus in GEASI only consider (4.35) for all computations.

As before, let $\phi_{\mathcal{E}}$ be the solution of the eikonal equation with given $\mathcal{E} = \{(\mathbf{x}_i, t_i)_{i=1}^N\}$. We define the region of influence as $\mathcal{R}_i \coloneqq \{\mathbf{x} \in \overline{\Omega} : \phi_{\mathcal{E}}(\mathbf{x}) = t_i + \delta(\mathbf{x}, \mathbf{x}_i)\}$. Furthermore, the derivatives of Φ_N with respect to $\mathbf{x}_i$ and t_i at $\mathbf{x} \in \mathring{\mathcal{R}}_i$ read as

$$\begin{aligned}\nabla_{\mathbf{x}_i}\Phi_N(\mathbf{x},\mathbf{x}_1,\dots,\mathbf{x}_N,t_1,\dots,t_N) &= \nabla_{\mathbf{x}_i}\delta(\mathbf{x}_i,\mathbf{x}),\\ \partial_{t_i}\Phi_N(\mathbf{x},\mathbf{x}_1,\dots,\mathbf{x}_N,t_1,\dots,t_N) &= 1,\end{aligned}$$

where we note that function is not differentiable on the boundary of the regions of interest. To compute the exponential map, we solve for each $i = 1,\dots,N$ the following initial value problem

$$\begin{cases}\dot{\gamma}_i(t) = -D(\gamma_i(t))\nabla\phi_{\{(\mathbf{x}_i,t_i)\}}(\gamma_i(t)),\\ \gamma_i(0) = \mathbf{x}\end{cases} \tag{4.37}$$

for $\mathbf{x} \in \Omega$. The regularity and boundedness of D and $\phi_{\{(\mathbf{x}_i,t_i)\}}$ already imply the existence of solutions. Then, we define the piecewise geodesic path γ as $\gamma(t) = \gamma_i(t)$ if $\gamma(t) \in \mathcal{R}_i$. Furthermore,

$$\bar{t} = \underset{t>0}{\arg\min}\{\gamma(t) \in \overline{B}_\zeta(\mathbf{x}_i) \text{ for } i = 1,\dots,N\} \tag{4.38}$$

is finite due to the assumptions regarding D for a small $\zeta > 0$. The inclusion of the ζ-balls essentially circumvent problems related to the non-differentiability of γ_i in the proximity of $(\mathbf{x}_i, t_i)$.

We note that by construction γ is a unit-speed geodesic for the length functional ((2.28) revisited). In this case, we define the exponential map in the direction $\bar{t}\dot{\gamma}(0)$ as $\operatorname{Exp}_{\mathbf{x}}(\bar{t}\dot{\gamma}(0)) = \gamma(\bar{t})$, and the logarithm $\operatorname{Log}_{\mathbf{x}}(\gamma(\bar{t})) = \bar{t}\dot{\gamma}(0)$ as its inverse. Note the the logarithm can efficiently be computed by tracking backward the geodesic from $\gamma(\bar{t})$ to $\mathbf{x}$.

Remark 4.3. *We note that points can belong to multiple regions of influence, at which the derivative of $\phi_{\mathcal{E}}$ might not be defined. However, due to the general functional-analytic setting the Lebesgue measure of these points is negligible.*

4.7.1.4 GEASI Algorithm

In this section, we introduce the GEASI Algorithm to solve (4.33) using a gradient-based approach. Here, we restrict to the specific functional

$$\mathcal{J}(\phi) := \int_\Gamma \frac{1}{2}(\phi(\mathbf{x}) - \widehat{\phi}(\mathbf{x}))^2 \,\mathrm{d}\mathbf{x},$$

where $\Gamma \subset \Omega$ is a subdomain of Ω with a positive Lebesgue measure and $\widehat{\phi} \in L^2(\Gamma,\mathbb{R})$ is a fixed square-integrable function. In numerical experiments, $\widehat{\phi}$ reflects the measurements on a known subdomain Γ, for which the quadratic mismatch on Γ between ϕ and $\widehat{\phi}$ with respect to the EASs is minimized. Examples of Γ include finite sets of points mimicking a contact recording map, the full endocardium/epicardium, or subregions of them.

According to Sections 4.7.1.2 and 4.7.1.3, the optimization problem for $N = 1$ simply reads

$$\min_{(\mathbf{x}_1,t_1)\in\mathcal{U}_1} \int_\Gamma \frac{1}{2}(t_1 + \delta(\mathbf{x}_1,\mathbf{x}) - \widehat{\phi}(\mathbf{x}))^2 \,\mathrm{d}\mathbf{x}. \tag{4.39}$$

To employ a gradient-based approach, we see that following Proposition 2 the gradient of $\mathcal{J}$ with respect to $\mathbf{x}_1$ simply reads as

$$\nabla_{\mathbf{x}_1}\mathcal{J}(\phi_{\{(\mathbf{x}_1,t_1)\}}) = -\int_\Gamma r(\mathbf{x},\mathbf{x}_1,t_1)\frac{D^{-1}(\mathbf{x}_1)\dot{\gamma}_{\mathbf{x}_1\to\mathbf{x}}(0)}{||\dot{\gamma}_{\mathbf{x}_1\to\mathbf{x}}(0)||_{D^{-1}(\mathbf{x}_1)}}\,\mathrm{d}\mathbf{x}, \tag{4.40}$$

where $\gamma_{\mathbf{x}_1\to\mathbf{x}}(t)$ is the geodesic path from $\mathbf{x}_1$ to $\mathbf{x}$ and $r(\mathbf{x},\mathbf{x}_1,t_1) = t_1 + \delta(\mathbf{x}_1,\mathbf{x}) - \widehat{\phi}(\mathbf{x})$ is the residual. Optimizing multiple points simultaneously yields an average direction weighted by the residuals r on Γ. Figure 4.21 depicts how the velocity field shown in Figure 4.20 translates to a descent direction to optimize (4.39).

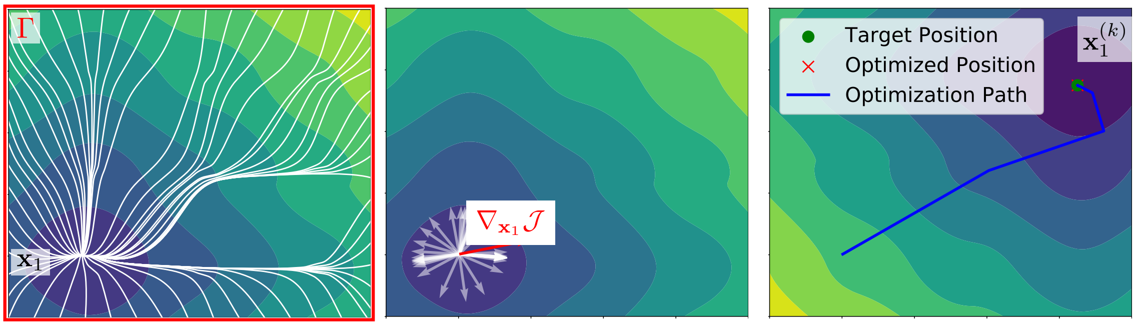


Figure 4.21: Visualization of the optimization problem in (4.39). Geodesics (white) originating from the single EAS $\mathbf{x}_1$ to distinct points on Γ (left) and corresponding gradients computed with (4.35) (middle). The highlighted direction (red) coincides with the gradient in (4.40). Right: by iteratively applying a gradient-based scheme we determine the optimal $(\mathbf{x}_1, t_1)$.

The extension to multiple EASs works similarly. A convenient formulation consists in splitting Γ into subdomains $\Gamma_i := \mathcal{R}_i \cap \Gamma$, each composed of those points activated by the EAS $\mathbf{x}_i$ (note that the set of points belonging to multiple regions Γ_i has Lebesgue measure 0). Then, the objective function reads as follows

$$\min_{\{(\mathbf{x}_i,t_i)_{i=1}^N\}\in\overline{\mathcal{U}_N}} \sum_{i=1}^N \int_{\Gamma_i} \frac{1}{2}(t_i + \delta(\mathbf{x}_i,\mathbf{x}) - \widehat{\phi}(\mathbf{x}))^2 \,\mathrm{d}\mathbf{x}. \tag{4.41}$$

Clearly, the optimization procedure for a single EAS readily translates to the case of multiple sites.

We found that similar to Section 4.5, a Gauss–Newton optimization proved beneficial to reduce the overall number of required optimization iterations, resulting in the following update rule

$$\begin{aligned}\mathcal{E}^{(k+1)} = & \operatorname*{arg\,min}_{\{(\mathbf{x}_i,t_i)_{i=1}^N\}\in\overline{\mathcal{U}_N}} \\ & \sum_{i=1}^{N} \frac{1}{2} \left\| \nabla_{\mathbf{x}_i,t_i} \mathcal{J}(\phi_{\mathcal{E}^{(k)}}) \begin{pmatrix} \mathbf{x}_i - \mathbf{x}_i^{(k)} \\ t_i - t_i^{(k)} \end{pmatrix} + \phi_{\mathcal{E}^{(k)}}(\mathbf{x}) - \widehat{\phi}(\mathbf{x}) \right\|_{L^2(\Gamma)}^2 . \end{aligned} \tag{4.42}$$

Here, $\mathcal{E}^{(k)} = \{(\mathbf{x}_i^{(k)}, t_i^{(k)})_{i=1}^N\}$ are the solutions of the previous iteration. To overcome local minima of the optimization problem (4.33) caused by non-unique solutions (see Remark 4.2) we additionally use an over-relaxation [26] with fixed $\beta_a = \frac{1}{\sqrt{2}}$. The resulting Algorithm 4.4 iteratively linearizes and solves the problem using the computed gradient from δ to match a given measured activation. We remark that the gradient properly reflects infinitesimal changes of activation times on $\mathcal{R}_i$ for each $\mathbf{x}_i$, but it is not capable of accurately capturing higher order effects like the change of $\mathcal{R}_i$. The experiments showed that rather than directly using $\mathcal{E}^{(k+1)}$ from (4.42) as the new solution, it is beneficial to take a step-size $\beta_s < 1$ and compute the convex combination of old and new solution according to this step size. For all experiments, we used $\beta_s = \frac{1}{2}$. For further details of the numerical realization we refer the reader to Section 4.7.3.

Algorithm 4.4: GEASI

```
Input  : initial $\mathbf{x}_i^{(0)}$ and $t_i^{(0)}$ defining $\mathcal{E}^{(0)} = \{(\mathbf{x}_i^{(0)}, t_i^{(0)})_{i=1}^N\}$,
         target activation $\widehat{\phi}(\mathbf{x})$ for $\mathbf{x} \in \Gamma$, conduction velocity tensor $D$
Output: optimal EASs $\mathbf{x}_i^*$ and times $t_i^*$
for $k = 1, \dots, K$ do
    $\widetilde{\mathcal{E}}^{(k)} = \mathcal{E}^{(k)} + \beta_a(\mathcal{E}^{(k)} - \mathcal{E}^{(k-1)})$
    solve the eikonal equation (4.29) for $\widetilde{\mathcal{E}}^{(k)} = \{(\widetilde{\mathbf{x}}_i^{(k)}, \widetilde{t}_i^{(k)})_{i=1}^N\}$
    compute all geodesics $\gamma_{\mathbf{x}_i \to \mathbf{x}}(t)$ for $\mathbf{x} \in \Gamma$ by solving (4.37)
    compute $\bar{\mathcal{E}}^{(k+1)}$ using (4.42) (with $\widetilde{\mathcal{E}}^{(k)}$)
    $\mathcal{E}^{(k+1)} = \widetilde{\mathcal{E}}^{(k)} + \beta_s \left(\bar{\mathcal{E}}^{(k+1)} - \widetilde{\mathcal{E}}^{(k)}\right)$
end
```

4.7.2 Extensions of GEASI

GEASI is a versatile optimization algorithm, which can be extended in several aspects. In this section, we focus on two such possible extensions. First, the topological gradient estimation allows for an accurate estimation of the number of EASs. Second, we modify the original objective function of GEASI to fit a given ECG.

4.7.2.1 Variable number of EASs: Topological Gradient

So far, we assumed the number of EASs N to be fixed. Since the optimal number of EASs is in general unknown, we subsequently propose a method to approximate the optimal N. As a possible approach to estimate N (which is not conducted in this work) one could start with a large number and successively remove distinct EASs that violate the constraint (4.32). However, this approach suffers from some major drawbacks:

- several local minima can occur leading to a strong dependency on the initial guess,
- enforcing (4.32) results in some numerical issues, e.g. dimension changes of the optimization problem and order of EAS removal.

In contrast, starting with a few (or even a single) EASs and subsequently introducing new EASs overcomes the above issues since according to Proposition 1 adding new sites does not increase the objective functional. In what follows, we briefly recall the topological gradient, which is used to compute the infinitesimal expansion of splitting a single EAS. This expansion is exploited to estimate the energy decrease of adding a new site.

Consider the case of a single EAS, i.e. $N = 1$. The topological gradient is defined as the effect on the solution of the associated eikonal equation if splitting a single EAS $\mathbf{x}_1$ into two new sites $\mathbf{x}_1 + \varepsilon\mathbf{n}$ and $\mathbf{x}_1 - \varepsilon\mathbf{n}$ in the direction of $\mathbf{n} \in \mathbb{S}^{d-1}$. We can directly infer from (4.30) that

$$\phi_{\mathcal{E}_\varepsilon}(\mathbf{x}) = \min\{t_1 + \delta(\mathbf{x}_1 - \varepsilon\mathbf{n}, \mathbf{x}), t_1 + \delta(\mathbf{x}_1 + \varepsilon\mathbf{n}, \mathbf{x})\}$$

for $\mathcal{E}_\varepsilon = \{(\mathbf{x}_1 + \varepsilon\mathbf{n}, t_1), (\mathbf{x}_1 - \varepsilon\mathbf{n}, t_1)\}$, where $\varepsilon > 0$ is sufficiently small. This topological operation divides the domain into two subdomains $\Omega_\varepsilon^- := \{\mathbf{x} \in \Omega : \delta(\mathbf{x}_1 - \varepsilon\mathbf{n}, \mathbf{x}) < \delta(\mathbf{x}_1 + \varepsilon\mathbf{n}, \mathbf{x})\}$ and $\Omega_\varepsilon^+ = \Omega \setminus \Omega_\varepsilon^-$. We can now expand ϕ_ε with respect to ε as follows

$$\begin{aligned}\phi_{\mathcal{E}_\varepsilon}(\mathbf{x}) &= t_1 + \delta(\mathbf{x}_1, \mathbf{x}) + \varepsilon\min\{-\nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1, \mathbf{x}) \cdot \mathbf{n}, \nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1, \mathbf{x}) \cdot \mathbf{n}\} + o(\varepsilon) \\ &= \Phi_1(\mathbf{x}, \mathbf{x}_1, t_1) - \varepsilon\,|\nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1, \mathbf{x}) \cdot \mathbf{n}| + o(\varepsilon),\end{aligned}$$

where Φ_1 was defined in (4.34) and we used (4.35). In this case, we call the quantity

$$j(\mathbf{x}, \mathbf{x}_1, \mathbf{n}) := |\nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1, \mathbf{x}) \cdot \mathbf{n}|\,. \tag{4.43}$$

the *topological gradient*. A visual example of the topological gradient is provided in Figure 4.22. We note that adding new optimal sites always decreases the objective functional unless $\nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1, \mathbf{x}) \cdot \mathbf{n} = 0$. Therefore, we shall define a criterion for adding a split. The decrease in energy of splitting a single site can be estimated as follows:

$$\nu_{S,\varepsilon} := \min_{\mathbf{n}\in\mathbb{S}^{d-1}} \int_\Gamma (\Phi_1(\mathbf{x}, \mathbf{x}_1, t_1) - \widehat{\phi}(\mathbf{x}))^2 - (\Phi_1(\mathbf{x}, \mathbf{x}_1, t_1) + \varepsilon j(\mathbf{x}, \mathbf{x}_1, \mathbf{n}) - \widehat{\phi}(\mathbf{x}))^2 \,\mathrm{d}\mathbf{x}. \tag{4.44}$$

Likewise, the effect of moving a source point in direction $\mathbf{n}$ is given by

$$\nu_{M,\varepsilon} := \min_{\mathbf{n}\in\mathbb{S}^{d-1}} \int_\Gamma (\Phi_1(\mathbf{x},\mathbf{x}_1,t_1) - \widehat{\phi}(\mathbf{x}))^2 \\ - (\Phi_1(\mathbf{x},\mathbf{x}_1,t_1) + \varepsilon\nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1,\mathbf{x})\cdot\mathbf{n} - \widehat{\phi}(\mathbf{x}))^2 \,\mathrm{d}\mathbf{x}.$$

The ratio $\frac{\nu_{M,\varepsilon}}{\nu_{S,\varepsilon}}$ has proven to be a robust score for adding new sites, which is verified in the numerical experiments. In particular, if the ratio is below a certain threshold, then a new EAS is introduced.

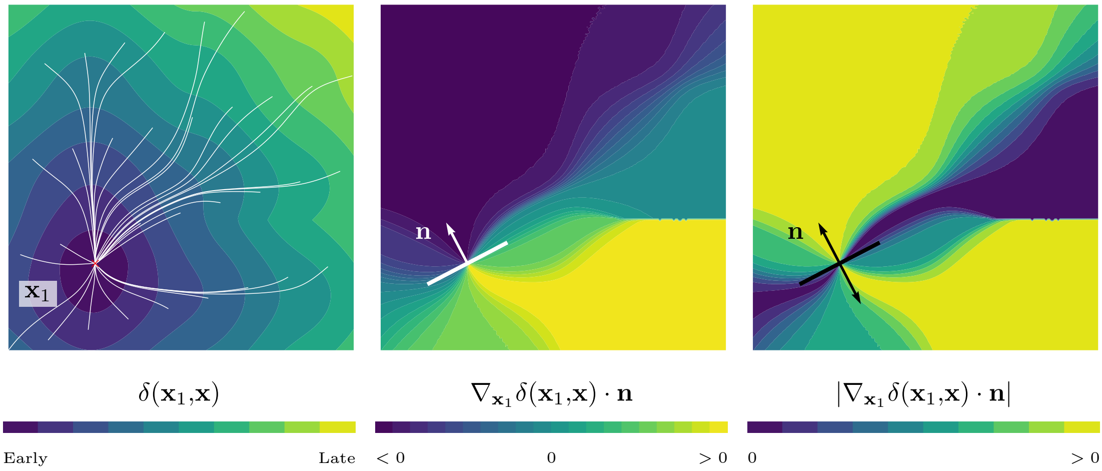


Figure 4.22: Left: geodesics (white) joining multiple points with $\mathbf{x}_1$. Contour plots of $\mathbf{x}\mapsto\nabla_{\mathbf{x}_1}\delta(\mathbf{x}_1,\mathbf{x})\cdot\mathbf{n}$ (middle) and $\mathbf{x}\mapsto j(\mathbf{x},\mathbf{x}_1,\mathbf{n})$ (right) for fixed $\mathbf{n}\in\mathbb{S}^{d-1}$. Moving a single EAS in the direction $\mathbf{n}$ alters the activation times $\delta(\mathbf{x}_1,\mathbf{x})$ as shown. In contrast, splitting in the same direction $\mathbf{n}$ is similar to simultaneously moving a source point in both directions, and keeping only the shorter geodesic (right).

4.7.2.2 Optimization using the ECG

The ECG is the observed signature of the electric activity of the heart, which is measured at selected locations on the chest. Being routinely acquired and non-invasive, the ECG is the ideal candidate for inferring cardiac activation in a clinical framework. Here, we will introduce a method to reconstruct the EASs directly from ECG measurements. To this end, we exploit the methods presented in [87, 95] to efficiently compute the ECG from activation maps of the eikonal equation. Finally, the quadratic mismatch of the computed and measured ECG is minimized, which yields optimal EASs.

From a modeling perspective, we denote $\Omega_T\subset\mathbb{R}^d\setminus\Omega$ as the whole body domain excluding the heart cavity $\Omega\subset\mathbb{R}^d$. The heart surface $\Gamma:=\overline{\Omega_T}\cap\overline{\Omega}$ is the interface between torso and heart, that is the boundary between the active myocardium and the rest of the body (for instance, endocardium plus epicardium). In this setting, $\Sigma:=\partial\Omega_T\setminus\Gamma$ is the chest, on which the aforementioned electrical signal is recorded. We denote by $\mathbb{T} \subset \mathbb{R}$ the considered time interval. A more detailed outline on the theory of lead fields was given in Section 3.3

An equation for the torso potential can be derived from bidomain theory for the cardiac tissue and the balance of currents in the body (see e.g. [51]). The resulting system of equations reads as follows:

$$\begin{cases} -\nabla \cdot (\mathbf{G}\nabla u_e) = \nabla \cdot (\mathbf{G}_i \nabla V_{\mathrm{m}}), & \text{in } \Omega \times \mathbb{T}, \\ -\nabla \cdot (\mathbf{G}_T \nabla u_T) = 0, & \text{in } \Omega_T \times \mathbb{T}, \\ -\mathbf{G}_T \nabla u_T \cdot \mathbf{n} = 0, & \text{in } \Sigma \times \mathbb{T}, \\ u_e(\mathbf{x}^-, t) = u_T(\mathbf{x}^+, t), & (\mathbf{x}, t) \in \Gamma \times \mathbb{T}, \\ \mathbf{G}_T(\mathbf{x}^+)\nabla u_T(\mathbf{x}^+, t) \cdot \mathbf{n} & \\ -\mathbf{G}(\mathbf{x}^-)\nabla u_e(\mathbf{x}^-, t) \cdot \mathbf{n} = \mathbf{G}_i(\mathbf{x}^-)\nabla V_{\mathrm{m}}(\mathbf{x}^-, t) \cdot \mathbf{n}, & (\mathbf{x}, t) \in \Gamma \times \mathbb{T}, \end{cases} \tag{4.45}$$

where the following quantities occur:

- $u_e \colon \Omega \times \mathbb{T} \to \mathbb{R}$ is the extracellular potential in the heart,
- $V_{\mathrm{m}} \colon \Omega \times \mathbb{T} \to \mathbb{R}$ is the transmembrane potential,
- $u_T \colon \Omega_T \times \mathbb{T} \to \mathbb{R}$ is the potential in the torso,
- $\mathbf{G}_T \colon \Omega_T \to S^d_{++}$ is the electric conductivity of the torso,
- $\mathbf{G}_i \colon \Omega \to S^d_{++}$ is the intracellular conductivity,
- $\mathbf{G}_e : \Omega \to S^d_{++}$ is the extracellular conductivity, and
- $\mathbf{G} = \mathbf{G}_i + \mathbf{G}_e$ is the bulk conductivity of the heart.

The normal vector $\mathbf{n}$ at $\mathbf{x} \in \Gamma$ points outwards, i.e. from the heart surface towards the torso, and is the outer normal vector for $\mathbf{x} \in \Sigma$. The points $\mathbf{x}^{\pm}$ associated with $\mathbf{x} \in \Gamma$ are obtained by taking the limit $\mathbf{x}^{\pm}_{\varepsilon} = \mathbf{x} \pm \varepsilon \mathbf{n}$ for $\varepsilon \to 0$.

The well-posedness of (4.45) follows from standard arguments for elliptic PDEs (see [57]). However, some care is required for the discontinuity across the heart-torso interface Γ. Let $\widetilde{\Omega} = \Omega \cup \Omega_T \cup \Gamma$ be the domain modeling the whole torso (including the heart), and

$$\widetilde{\mathbf{G}} = \begin{cases} \mathbf{G}, & \text{in } \Omega, \\ \mathbf{G}_T, & \text{in } \Omega_T, \end{cases} \qquad \widetilde{u} = \begin{cases} u_e, & \text{in } \Omega \times \mathbb{T}, \\ u_T, & \text{in } \Omega_T \times \mathbb{T}. \end{cases}$$

Following [51], we assume that

1. $\Omega, \Omega_T \subset \mathbb{R}^d$ are Lipschitz domains,
2. $\mathbf{G}_i, \mathbf{G}_e \in C^1(\Omega, S^d_{++})$ and $\widetilde{\mathbf{G}} \in L^\infty(\widetilde{\Omega}, S^d_{++})$,
3. $V_{\mathrm{m}}(\cdot, t) \in W^{2,p}(\Omega)$ for all $t \in \mathbb{T}$ with $p > d$.

Proposition 3. *Under the above assumptions, the weak formulation of* (4.45) *given by*

$$\text{find } \widetilde{u} \in H^1(\widetilde{\Omega}) \text{ s.t. } \int_{\widetilde{\Omega}} \widetilde{\mathbf{G}} \nabla \widetilde{u} \cdot \nabla v \, \mathrm{d}\mathbf{x} = - \int_{\Omega} \mathbf{G}_i \nabla V_m \cdot \nabla v \, \mathrm{d}\mathbf{x} \quad \forall v \in H^1(\widetilde{\Omega}) \tag{4.46}$$

is well-defined. In particular, there exists a unique solution up to an additive constant (the reference potential).

The proof directly follows from the Lax–Milgram theorem [57] by noting that $V_{\mathrm{m}}(\cdot, t) \in W^{2,p}(\Omega)$ and $\widetilde{u}(\cdot, t) \in W^{1,p}(\widetilde{\Omega})$ for $p > d$ and all $t \in \mathbb{T}$. In such a manner, all the lead fields between all necessary electrode pairs of a clinically acquired ECG can be computed.

Solving (4.45) is numerically costly for the standard 12-lead ECG, since we only evaluate u_T at selected locations. Note that the system must be solved for every $t \in \mathbb{T}$. Thus, we adopt the integral representation, introduced in Section 3.3:

$$V_l(t) = \int_{\Omega} \mathbf{G}_i(\mathbf{x}) \nabla V_{\mathrm{m}}(\mathbf{x}, t) \cdot \nabla Z_l(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag*{((3.18) revisited)}$$

where $Z_l \colon \widetilde{\Omega} \to \mathbb{R}$ are the lead fields, outlined in Section 3.3.

Next, we compute the torso potential u_T in (4.45) from the solution of the anisotropic eikonal equation (4.29). To this end, we assign the transmembrane potential $\widetilde{V_{\mathrm{m}}}$ accordingly to a fixed waveform $U \colon \mathbb{R} \to \mathbb{R}$ shifted by the activation time $\phi_{\mathcal{E}}$ as follows

$$\widetilde{V_{\mathrm{m}}}(\mathbf{x}, t, \mathcal{E}) = U(t - \phi_{\mathcal{E}}(\mathbf{x})).$$

In Section 3.1, we discussed the origins of the electrical activation in the heart and have shown that for traveling front, the bistable equation results in a parameterizable continuous waveform. Assuming an additional repolarization, we can define the waveform in terms of physiological parameters:

$$U(\xi) = K_0 + \frac{K_1 - K_0}{2} \left[\tanh\left(2\frac{\xi}{\tau_1}\right) - \tanh\left(2\frac{\xi - \mathrm{APD}}{\tau_2}\right) \right], \tag*{((3.6) revisited)}$$

which is visualized in Figure 4.23.

Furthermore, the conduction velocity tensor D in the anisotropic eikonal equation in (4.29) is linked to the electric conductivity as follows:

$$D = \frac{\alpha^2}{\beta} \mathbf{G}_e \mathbf{G}^{-1} \mathbf{G}_i,$$

where β is the surface-to-volume ratio and α is a rescaling factor either experimentally estimated or obtained by solving the monodomain equation in a cable propagation setup [95]. Note that in all conducted experiments we assumed an equal anisotropy ratio $\mathbf{G}_i = \lambda \mathbf{G}_e$, from which $D = \frac{\alpha^2}{\beta} \frac{\lambda}{1+\lambda} \mathbf{G}_i$ follows. All parameters adopted in this study are provided in Table 4.6.

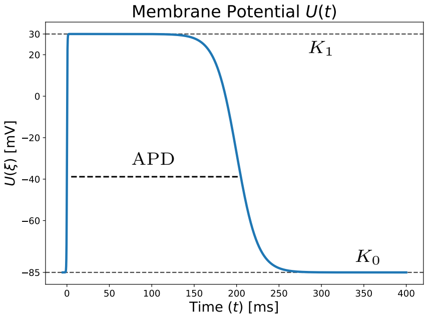


Figure 4.23: Membrane voltage waveform as a function of time, equivalent to (3.6) with parameters from Table 4.6. The continuous formulation allows for an analytical derivation in (4.47).

Parameter	Description	Value	Unit
t	time	$[0, T]$	ms
λ	equal anisotropy ratio	3	
$\mathbf{G}_T$	torso conductivity	0.2	$\frac{\text{mS}}{\text{mm}}$
α	conduction velocity scaling	20	$\text{mm}(\text{ms mS}^{-1/2})^{-1}$
β	surface-to-volume ratio	10^3	mm^{-1}
K_0	resting potential	-85	mV
K_1	plateau potential	30	mV
τ_1	depolarization time-scale	1	ms
τ_2	repolarization time-scale	50	ms
APD	action potential duration	200	ms

Table 4.6: Parameters to compute the ECG from the eikonal solution $\phi_{\mathcal{E}}$.

We emphasize that $\phi_{\mathcal{E}} \in C^{0,1}(\overline{\Omega})$ only implies $\widetilde{V_{\mathrm{m}}}(\cdot, t, \mathcal{E}) \in W^{1,p}(\Omega)$ and not $\widetilde{V_{\mathrm{m}}}(\cdot, t, \mathcal{E}) \in W^{2,p}(\Omega)$ as required for $t \in \mathbb{T}$ and $\mathcal{E} \in \mathcal{U}_N$. However, the aforementioned theory is still valid in this case with some major modifications that are beyond the scope of this work. Again, we refer to [51] and the references therein for further details.

In what follows, we intend to compute the sensitivities of the ECG with respect to the parameter set $\mathcal{E} \in \mathcal{U}_N$. In the problem, only the activation map $\phi_{\mathcal{E}}$ appearing in the definition of $\widetilde{V_{\mathrm{m}}}$ depends on the parameters in $\mathcal{E}$. Note that the chain rule straightforwardly implies

$$\nabla_{\mathcal{E}} \widetilde{V_{\mathrm{m}}} = -\frac{\partial U}{\partial \xi} \nabla_{\mathcal{E}} \phi_{\mathcal{E}}.$$

The use of the aforementioned smooth waveform allows for a continuous analytical derivative $\frac{\partial U}{\partial \xi}$. Details on the derivation of the term $\nabla_{\mathcal{E}} \phi_{\mathcal{E}}$ were already given in Section 4.7.1.3.

Then, the derivative $\nabla_{\mathcal{E}} V_l$ is computed from (3.19) and reads as

$$\nabla_{\mathcal{E}} V_l(t,\mathcal{E}) = \int_\Omega \Big(\mathbf{G}_i(\mathbf{x})\nabla^2_{\mathcal{E},\mathbf{x}} \widetilde{V_\mathrm{m}}(\mathbf{x},t,\mathcal{E})\Big)\nabla Z_l(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{4.47}$$

Finally, in this model the set of EASs $\mathcal{E}$ is computed from the measured ECG $\widehat{V}_l : I \to \mathbb{R}$ as follows:

$$\min_{\mathcal{E}\in\overline{\mathcal{U}_N}} \frac{1}{2} \sum_{l=1}^{L} \int_{\mathbb{T}} \Big(V_l(t,\mathcal{E}) - \widehat{V}_l(t)\Big)^2 \mathrm{d}t, \tag{4.48}$$

which is solved using the Gauss–Newton algorithm in a similar fashion to Algorithm 4.4. In particular, the update of the set $\mathcal{E}$ reads as follows

$$\begin{aligned} \mathcal{E}^{(k+1)} = & \operatorname*{arg\,min}_{\{(\mathbf{x}_i,t_i)_{i=1}^N\}\in\overline{\mathcal{U}_N}} \\ & \sum_{l=1}^{L}\sum_{i=1}^{N} \frac{1}{2} \Big|\Big|\nabla_{\mathbf{x}_i,t_i}\mathcal{J}(\mathcal{E}^{(k)})(\mathbf{x}_i-\mathbf{x}_i^{(k)}, t_i - t_i^{(k)})^\top + V_l(t,\mathcal{E}^{(k)}) - \widehat{V}(t)\Big|\Big|^2_{L^2(\mathbb{T})} \end{aligned} \tag{4.49}$$

with the modified objective functional

$$\mathcal{J}(\mathcal{E}) = \frac{1}{2} \sum_{l=1}^{L} \int_{\mathbb{T}} \big(V_l(t,\mathcal{E}) - \widehat{V}_l(t)\big)^2 \mathrm{d}t.$$

The numerical integration in (4.49) is realized using the trapezoidal rule.

Remark 4.4. *There are several numerical issues related to the optimization:*

1. *The waveform (3.6) is a rough approximation of a physiological action potential modelled the electrophysiology of a cell. The function U and the scaling parameter α may be simultaneously approximated from a generic ionic model by solving a 1-D propagation in a (possibly very long) cable with uniform coefficients. Alternatively, it is possible to show that (U, U', α) solves a nonlinear eigenvalue problem involving the ionic model [77].*
2. *The approximation $\widetilde{V_m}$ is not suitable to model the repolarization of the heart which is responsible for the T-wave. The reason is that the polarity of the T-wave, in general in accordance with the polarity of the QRS complex, can only arise from a heterogeneity in the action potential. Such heterogeneity might be introduced here, but it would be hard to reproduce the smoothing effect due to diffusion currents. Finally, the eikonal model is not suitable for the repolarization because, opposed to the depolarization phase, the repolarization front is of the same order of the size of the domain, impeding a proper perturbation analysis. In this work, the repolarization time is $\phi_{\mathcal{E}}(\mathbf{x}) + APD$, hence it satisfies the same equation as $\phi_{\mathcal{E}}$, but with a shifted time.*

3. *Equation (4.47) requires higher order derivatives of $\widetilde{V_m}$ and subsequently $\phi_{\mathcal{E}}$. While we computed the derivative $\nabla_{\mathcal{E}}\widetilde{V_m}$ as previously discussed, the computation of $\nabla_{\mathbf{x}}\widetilde{V_m}$ is numerically achieved on the reference element.*
4. *It is important to mention that the gradient computation for the minimization of (4.48) is usually much more costly compared to optimizing the problem in the eikonal formulation from (4.39), since the size of Γ is much smaller compared to Ω. However, to compute $\nabla_{\mathbf{x}_i,t_i}\mathcal{J}$ we need the activation times and their derivatives in Ω, which necessitates the computation of the geodesics from each point of our domain to the EAS $\mathbf{x}_i$. The computational complexity is significantly larger than the complexity for (4.39). Further strategies to reduce additional computational costs are presented in Section 4.7.5.*

4.7.3 Discretization

In this section, we elaborate on the discretization aspects for Algorithm 4.4, which encompasses the steps: over-relaxation of $\mathcal{E}^{(k+1)}$, solving the eikonal equation, computation of the geodesics and update of $\mathcal{E}^{(k+1)}$.

4.7.3.1 Solving the Eikonal Equation

The discrete function space for the eikonal equation is the space of volumetric Lagrange $\mathcal{P}^1$-finite elements defined on triangular ($d = 2$) and tetrahedral ($d = 3$) meshes discretizing Ω, respectively. Moreover, the discrete measurements in Γ are degrees of freedom (DOFs) of the mesh.

Typically, finite element solvers require the initiation sites to coincide with DOFs of the mesh. However, since the original problem (4.33) expresses $\mathbf{x}_i$ as a continuous quantity, we identify the DOFs of the actual element containing the activation site. Then, these DOFs are added to the Dirichlet boundary Γ_D with fixed activation times given by $t_i + ||\mathbf{x} - \mathbf{x}_i||_{D^{-1}(\mathbf{x}_i)}$ for $\mathbf{x} \in \Gamma_D$ due to structural assumptions regarding the $\mathcal{P}^1$-finite element space. For the rare case of two or more initiation sites residing in the same element, we use the properties of (4.30) and (4.32) to compute the activation times.

4.7.3.2 Computation of Geodesics

In this work, we employ Heun's method (second order explicit Runge–Kutta scheme) to solve (4.37), which proved to be stable and efficient in numerical experiments. Due to the convergence of the ODE system to a stable node $\mathbf{x}_i$ we terminate the iteration if the ℓ^2-norm of two consecutive iterations is below 10^{-10}. In practice, the ODE system is solved independently on each region of interest $\mathcal{R}_i$ incorporating the whole set $\mathcal{E}$. Since the gradient of the eikonal solution for the chosen discretization is a $\mathcal{P}^0$-finite element function (i.e. piecewise constant), we advocate a standard L^2-projection onto the $\mathcal{P}^1$-finite element space [82]. Note that this projection can be realized by solving a linear system involving the mass matrix in $\mathcal{P}^1$. Since the boundary of Ω is in general curved, we project the geodesics back onto $\partial\Omega$ if they are outside of the domain after each update.

As remarked in Section 4.7.1.3, the gradient of the eikonal solution is discontinuous around each $\mathbf{x}_i$. To enforce regular gradients at each $\mathbf{x}_i$ after the L^2-projection of the previous eikonal solution $\widetilde{\phi}_{\mathcal{E}}$, we recompute the points with vanishing gradient by the subsequent variational problem with Tikhonov regularization for $c \in [\![d]\!]$ and balancing parameter $\lambda > 0$ as follows:

$$\begin{pmatrix} [\nabla\phi_{\mathcal{E}}(\mathbf{y}_1)]_c \\ \vdots \\ [\nabla\phi_{\mathcal{E}}(\mathbf{y}_{d+1})]_c \end{pmatrix} = \operatorname*{arg\,min}_{\mathbf{n}\in\mathbb{R}^{d+1}} \frac{1}{2}\langle \mathbf{\Psi}(\mathbf{x}_i), \mathbf{n}\rangle^2 + \frac{\lambda}{2}\left\|\mathbf{n} - \begin{pmatrix} [\nabla\widetilde{\phi}_{\mathcal{E}}(\mathbf{y}_1)]_c \\ \vdots \\ [\nabla\widetilde{\phi}_{\mathcal{E}}(\mathbf{y}_{d+1})]_c \end{pmatrix}\right\|^2 .$$

Here, $\mathbf{\Psi} = (\psi_1, \ldots, \psi_{d+1})^\top$ and $\{\mathbf{y}_j\}_{j=1}^{d+1}$ are the collections of $\mathcal{P}^1$-basis functions and degrees of freedom associated with the element containing $\mathbf{x}_i$, respectively. Figure 4.24 depicts the effect of this regularization on the solution around $\mathbf{x}_i$.

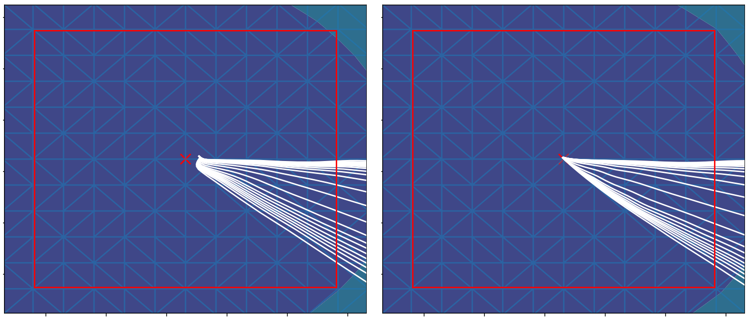


Figure 4.24: Left: zoom of family of geodesics emanating from a single EAS without special handling. Note that before the L^2-projection of $\nabla\phi_{\mathcal{E}}$ geodesics are not guaranteed to reach the EAS. Right: after the L^2-projection all geodesic curves actually reach the EAS.

The gradient computation in (4.35) is also sensitive to the choice of the step sizes of (4.37), which we choose as $5 \cdot 10^{-2}h$ with h being the average element size. As already described in Section 4.7.1.3, we compute the geodesic direction not directly at $\mathbf{x}_i$, but rather in a small ζ-neighborhood with $\zeta = 0.5h$ as advocated in (4.38). Numerically, the convergence of geodesics to this neighborhood is not ensured and non-converged geodesics (rarely occurring) do not affect the optimization.

4.7.3.3 Update of $\mathcal{E}^{(k+1)}$

Next, we optimize (4.42), where we have to ensure $\mathcal{E} \in \overline{\mathcal{U}_N}$. The constraint $\mathbf{x}_i \in \overline{\Omega}$ is mesh-dependant and allows for no general analytical solution, potentially limiting the available optimization implementations. To overcome this hurdle, we again use the Moreau envelope of Example 2.10 in a proximal point algorithm enforcing $\mathcal{E} \in \overline{\mathcal{U}_N}$. The integration is realized using an exact simplex quadrature rule.

The metric of the Moreau envelope for (4.42) is given by $\mathbf{M}_{i,\mathcal{E}^{(k)}} := \frac{1}{\tau}\mathbf{I} - \mathbf{J}^\top_{i,\mathcal{E}^{(k)}}\mathbf{J}_{i,\mathcal{E}^{(k)}}$ for $\tau < \|\mathbf{J}^\top_{i,\mathcal{E}^{(k)}}\mathbf{J}_{i,\mathcal{E}^{(k)}}\|^{-1}$, where $\mathbf{J}_{i,\mathcal{E}^{(k)}} := \nabla_{(\mathbf{x}_i,t_i)}\mathcal{J}(\mathbf{x}_i^{(k)}, t_i^{(k)})$. Thus, the Moreau envelope reads as

$$f(\overline{\mathcal{E}}) := \min_{\mathcal{E}\in\overline{\mathcal{U}_N}} \sum_{i=1}^{N} \frac{1}{2} \left\| \mathbf{J}_{i,\mathcal{E}^{(k)}} \begin{pmatrix} \mathbf{x}_i - \mathbf{x}_i^{(k)} \\ t_i - t_i^{(k)} \end{pmatrix} + \mathbf{r}_{\mathcal{E}^{(k)}}(\mathbf{x}) \right\|^2_{L^2(\Gamma)} + \frac{1}{2} \left\| \begin{pmatrix} \overline{\mathbf{x}}_i - \mathbf{x}_i \\ \bar{t}_i - t_i \end{pmatrix} \right\|^2_{\mathbf{M}_{i,\mathcal{E}^{(k)}}} \tag{4.50}$$

with $\mathcal{E} = \{(\mathbf{x}_i, t_i)_{i=1}^N\}$ and $\overline{\mathcal{E}} = \{(\overline{\mathbf{x}}_i, \bar{t}_i)_{i=1}^N\}$, and $\mathbf{r}_{\mathcal{E}^{(k)}}(\mathbf{x}) = \phi_{\mathcal{E}^{(k)}}(\mathbf{x}) - \widehat{\phi}(\mathbf{x})$. As already discussed in Example 2.10, this particular choice of the metric [26] allows for an explicit solution to (4.50) given the projection onto $\mathcal{U}_N$. This method is usually referred to as ISTA [44]. In summary, the iteration step of the proximal point algorithm reads as

$$\widehat{\mathcal{E}} = \left\{ \left(\operatorname*{proj}_{\mathcal{U}_1} \left(\begin{pmatrix} \overline{\mathbf{x}}_i \\ \bar{t}_i \end{pmatrix} - \int_\Gamma \tau \mathbf{J}^\top_{i,\mathcal{E}^{(k)}} \left(\mathbf{J}_{i,\mathcal{E}^{(k)}} \begin{pmatrix} \overline{\mathbf{x}}_i - \mathbf{x}_i^{(k)} \\ \bar{t}_i - t_i^{(k)} \end{pmatrix} + \mathbf{r}_{\mathcal{E}^{(k)}}(\mathbf{x}) \right) \mathrm{d}\mathbf{x} \right) \right)_{i=1}^{N} \right\}.$$

Thus, the resulting optimal set is $\widehat{\mathcal{E}} = \{(\widehat{\mathbf{x}}_i, \widehat{t}_i)_{i=1}^N\}$. Note that the convexity of the projection depends on the convexity of the domain. In practice, hardly any cardiac mesh is convex, but nevertheless the proposed method generated reliable results for sufficiently small step sizes. The gradient direction of the Moreau-envelope is

$$\nabla f_{\mathbf{x}_i,t_i}(\overline{\mathcal{E}}) = \tau^{-1} \begin{pmatrix} \overline{\mathbf{x}}_i - \widehat{\mathbf{x}}_i \\ \bar{t}_i - \widehat{t}_i \end{pmatrix}$$

In this case, the unconstrained problem is solved using L-BFGS [21].

4.7.4 Numerical Results

Next, we present numerical results for various methods discussed above, where we focus on four setups to test GEASI on theoretical and cardiac problems:

$\boxed{\mathbf{1}}$ The square domain presented in Figure 4.20 with a periodic conduction velocity field, where the measurement domain Γ coincides with the boundary of the surface.

$\boxed{\mathbf{2}}$ On a simplified 2D left ventricle (LV)-slice geometry with a transmural fiber rotation. Fiber and transverse velocities were set 0.6 and 0.4 $\frac{\mathrm{m}}{\mathrm{s}}$, respectively. The measurement domain Γ is the outer ring of the domain, i.e. an epicardial slice.

$\boxed{\mathbf{3}}$ On a clinically sampled, endocardial electrical mapping, measured during cardiac resynchronization therapy (CRT). The measurements were projected onto a patient-specific LV heart geometry and acquired by MRI. We mapped fiber orientations into the model using the approach described in [9]. Fiber, transverse and cross conduction velocities were set 0.6, 0.4 and 0.2 $\frac{\mathrm{m}}{\mathrm{s}}$, respectively.

$\boxed{\mathbf{4}}$ A full biventricular, trifascicular LV/RV human heart geometry with 1000 measurement points Γ distributed evenly along the epicardium. Fiber orientations were mapped as before. To account for the non-existent Purkinje fiber tree, an additional fast-conducting isotropic endocardial layer with a propagation velocity of 1.5 $\frac{\mathrm{m}}{\mathrm{s}}$ is used.

Further numerical specifications of the aforementioned setups are listed in Table 4.7. A summary of the example setups is provided in Figure 4.25.

	DOFs	Size [cm^d]	h [mm]	Runtime [h]	Topological gradient	ECG
$\boxed{\mathbf{1}}$	50^2	$2 \cdot 2$	0.4	1/2 (ECG)	✓	✓
$\boxed{\mathbf{2}}$	7980	$2 \cdot 2$	0.11	1/2 (ECG)	✓	✓
$\boxed{\mathbf{3}}$	$1.5 \cdot 10^4$	$10.7 \cdot 8.9 \cdot 9.5$	1.9	2.5	✓	
$\boxed{\mathbf{4}}$	$1.08 \cdot 10^5$	$10.3 \cdot 8.1 \cdot 12.6$	0.66	3.5/18 (ECG)		✓

Table 4.7: Selected parameters for each setup. The size refers to the bounding box of the setups and h is the average element spacing. Tested extensions are indicated by check marks. The ECG runtimes are separately denoted behind the dash as the experiments are more computationally demanding.

In this work, we use a custom `C++` implementation of the FIM to solve (4.29). Note however that the method is independent of the chosen eikonal solver and may benefit from higher order or smoother solutions of different solvers. A minimal working example for the method can be found on GitHub[2], but is limited to the isotropic eikonal equation on structured grids using the Fast Marching Method [112, 118] without FEM. The ECG and geodesic computations (see (3.18) and (4.37), respectively) and its Jacobian computation are calculated using the Tensorflow framework[3], making use of available GPUs and enabling automatic derivation of V_l with respect to $\widetilde{V_{\mathrm{m}}}$. All computations were performed on a single desktop machine with an Intel Core i7-5820K CPU using 6 cores of each 3.30GHz, 32GB of working memory and a NVidia RTX 2080 GPU.

4.7.4.1 Activation Time Optimization

In Figure 4.26, we present the results of GEASI for the 2D experiments: In the first iterations of the square example, the EASs are moved to the center of the domain to promote a good overall fit during optimization. As the sites approach the center, fine

[2] `https://github.com/thomgrand/geasi_grid_demo`

[3] `https://www.tensorflow.org/`

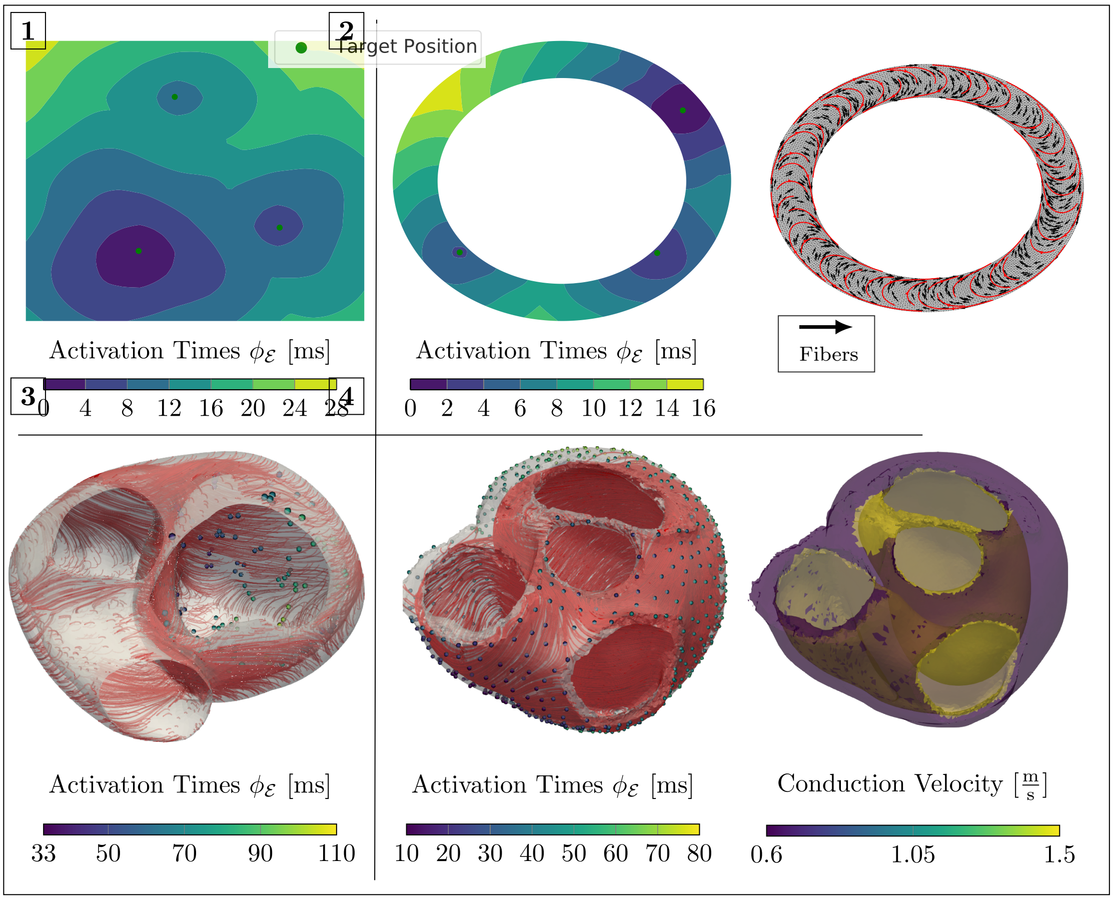

1
2
Target Position
Activation Times $\phi_\mathcal{E}$ [ms]
0 4 8 12 16 20 24 28
Activation Times $\phi_\mathcal{E}$ [ms]
0 2 4 6 8 10 12 14 16
Fibers
3
4
Activation Times $\phi_\mathcal{E}$ [ms]
33 50 70 90 110
Activation Times $\phi_\mathcal{E}$ [ms]
10 20 30 40 50 60 70 80
Conduction Velocity $[\frac{\text{m}}{\text{s}}]$
0.6 1.05 1.5

Figure 4.25: Activation times for all setups considered along with fiber orientations (if available). The isotropic conduction velocity of $\boxed{\textbf{1}}$ is presented in Figure 4.20 and exhibits no fiber orientation due to isotropy. Note that $\boxed{\textbf{3}}$ was measured in-vivo and thus no ground truth is available.

details on the boundary can be fitted by minimizing the mismatch defining the optimal points. The idealized LV model additionally requires a non-convex projection since the fiber alignment favors movements on the endocardial wall. The optimization still works for this case, even though the problem in (4.50) becomes non-convex.

Next, we concentrate on 3D experiments in Figures 4.27 and 4.28, for which we alter the number of initiation points for both models. Even though we can not ensure that the activation of the clinically acquired CRT patient can be described by the eikonal model with the simple rule-based fiber orientation, the results on the CRT masurements provide an overall low root-mean-square error (RMSE) between modelled and measured activation times. In the presence of a single EAS, the fit is (expectedly) sub-optimal since the activation requires a more complex activation pattern. With three or more EASs, we get a much better fit, evenly distributed throughout the ventricles, but additional

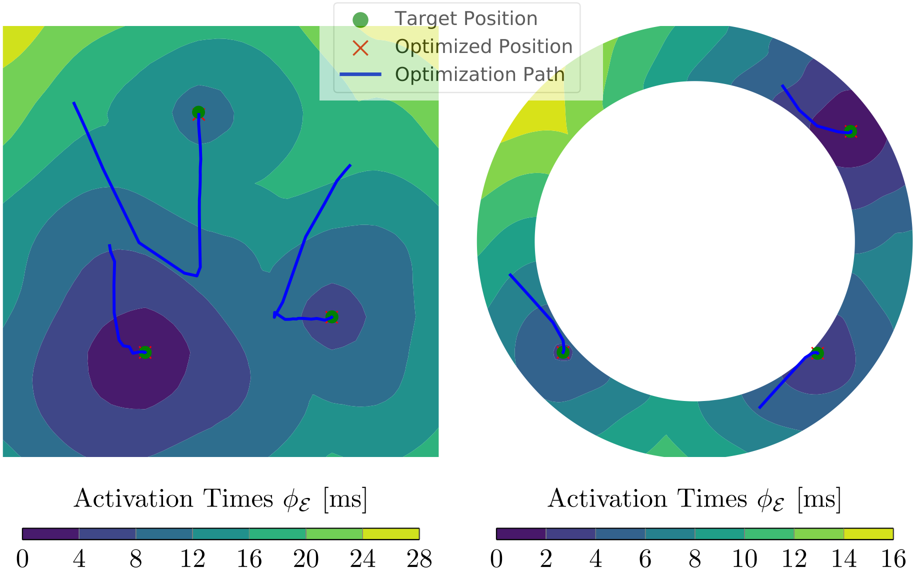


Figure 4.26: Results of GEASI for both 2D experiments. We located the exact initiation sites with only a few iterations, both for the heterogeneous velocity case (left) and in the presence of non-convex projections for the idealized LV model (right).

initiation sites are moved from the septum to the LV. These initiation sites seem to be an unrealistic consequence of the simplified modeling of fiber directions. We can achieve even better results by sucessively increasing the number of initiation sites, but this only reveals the nature of the ill-posed problem: By increasing the complexity of our model, we can more closely approximate the presented activation map (cf. Proposition 1).

The trifascicular model has a higher resolution in comparison to the CRT model with an added fast conducting sub-endocardial layer, which is utilized in all longer geodesics from the measurements to the initiation sites. For this reason, it is important to properly project the geodesics in each iteration on the endocardium in a fast way. For further details of the actual implementation we refer the reader to Section 4.7.5. As a result, when using less EASs than in the ground truth we already achieve convincing numerical results, which is visualized in Figure 4.28. If we incorporate 6 initiation sites, we get a very good fit, even though one of the activation sites is deactivated before convergence due to (4.32). The three septal points are jointly modelled by two EASs accounting for the deactivated point. Adding points beyond the given ones did not yield any improvement as they are deactivated by other points during the optimization (not shown).

4.7.4.2 Topological Gradient

We also tried to estimate the correct number of EASs by using topological gradients (see Section 4.7.2.1), where we analyzed all 2D setups and the CRT patient. In this case,

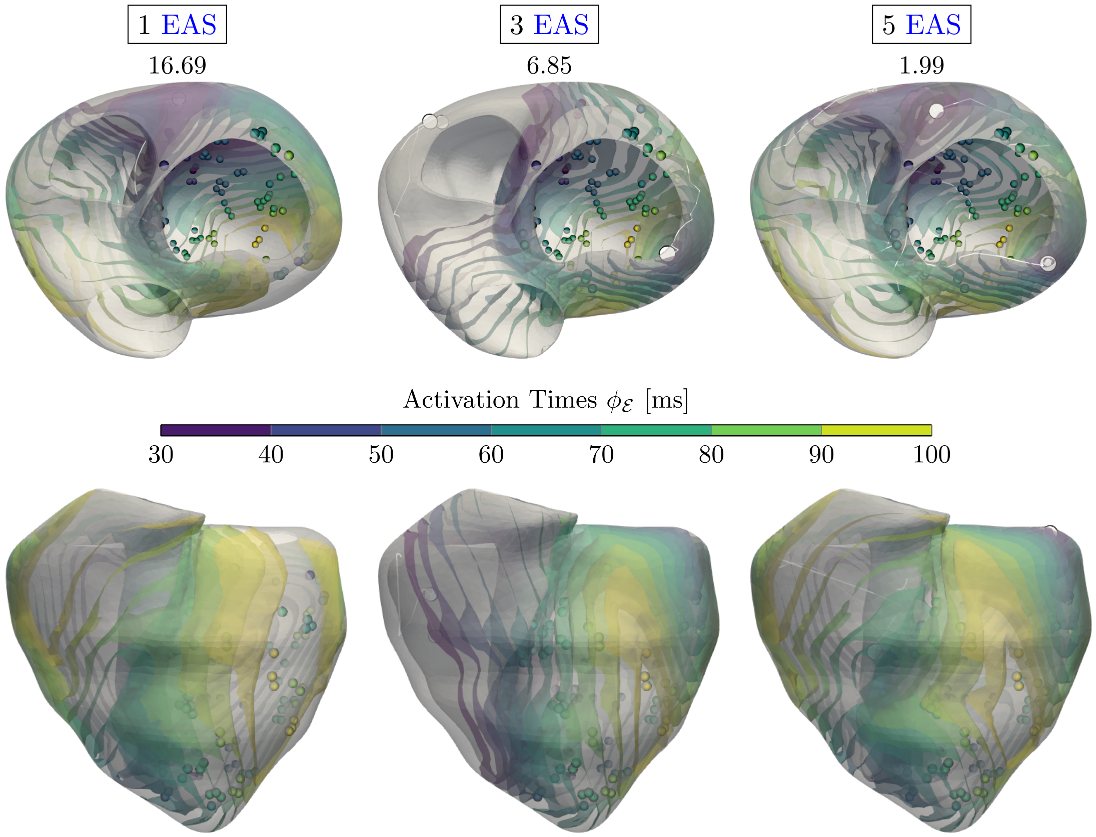


Figure 4.27: Results for the CRT experiment with varying number N of EASs along with the RMSE (in ms) shown above each experiment. The color-coded spheres indicate the observed activation times, while the white circles represent the optimized EAS positions. The white trailing paths show the optimization path over the iterations. Increasing N lowers the overall RMSE, but may result in physiologically unlikely EAS (e.g. top of the left ventricle for $N = 5$).

a splitting can only occur if the ratio $\frac{\nu_{M,\varepsilon}}{\nu_{S,\varepsilon}}$ is below 10^{-1} (2D)/$2.5 \cdot 10^{-1}$ (3D) and the maximum Euclidean distance of the position of two consecutive iterates among all EASs is smaller than $10^{-2}h$ (with h being the average element spacing).

The minimizer of (4.44) is chosen by evaluating 360 (2D)/5625 (3D) directions, which are evenly distributed on the hypersphere. We additionally ensure that the splitting direction is feasible (i.e. it does not point outside the domain) by projecting the samples onto the mesh. To avoid two coalescing EASs inside one element after a split, the points are moved apart by $2h$ from the original site.

In Figure 4.29, we collected the results for all 2D experiments using this method and plot the ratio $\frac{\nu_{M,\varepsilon}}{\nu_{S,\varepsilon}}$ over the iterations. The first EAS is moved towards the center of the ground truth EASs, and subsequently several splits occur that closely match the ground truth sites. A similar behavior can be observed in the idealized LV model.

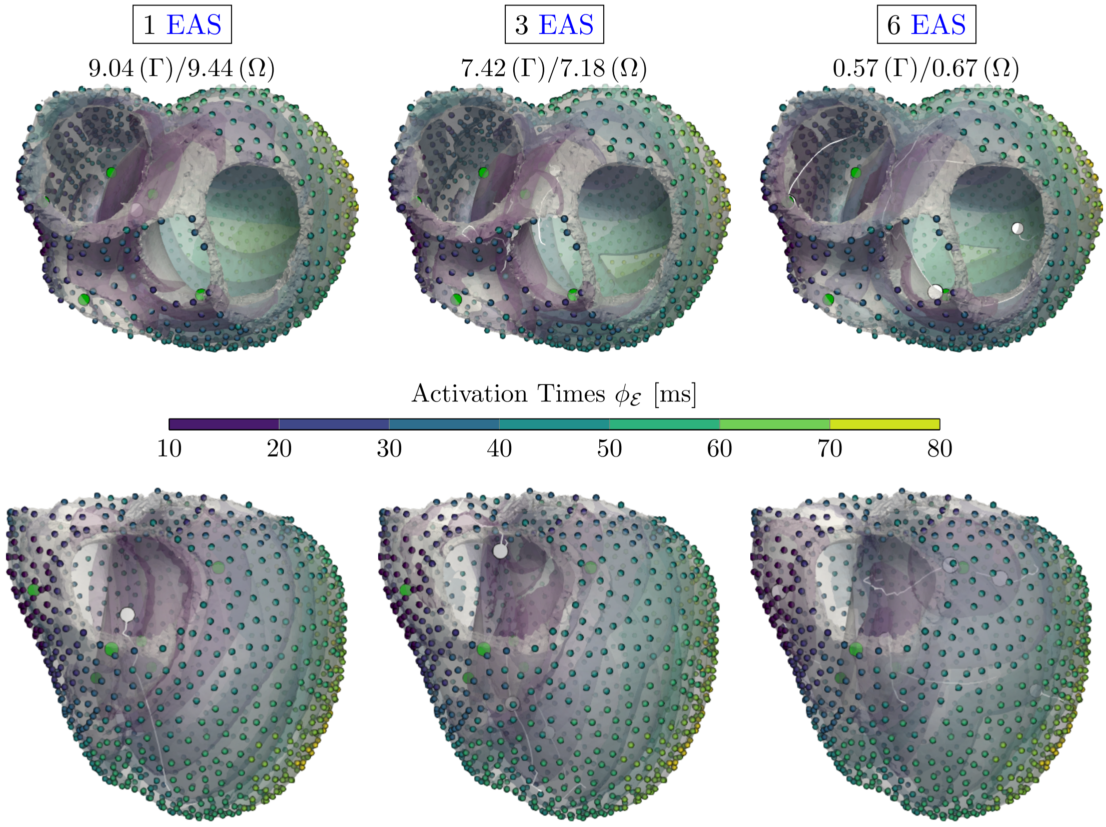


Figure 4.28: Results for the trifascicular experiment along with the RMSE (in ms) shown above the experiments for both Γ and Ω. The color-coded spheres indicate the observed activation times. The white and green circles represent the optimized and target EAS, respectively. The overall RMSE activation error is very low if using the correct number of initiation sites ($N = 6$), but we already obtain a good fit with fewer sites.

For the CRT patient in Figure 4.30, neither the ground truth EASs nor the fiber distribution and velocities in Ω are known. In total, the algorithm introduced 8 splits (i.e. $N = 9$), of which 4 are deactivated during optimization since they violate (4.32). Only those final EASs are shown in the right plot of Figure 4.30. Moreover, we can see that three main clusters are identified, where one initiation cluster is located at the upper part of the anterior septum. The optimization in this region is further complicated by the very thin wall of the 3D mesh, which likely causes the high number of splits. We highlight that constant (in time) split ratios are caused by temporarily deactivated EASs violating (4.32). To conclude, we get a tremendous fit with the presented measurement points despite the aforementioned model assumptions. Moreover, the topological gradient could be successfully applied to all 2D models leading to the correct estimate for the number of EASs and also matching the correct sites. The corresponding results in 3D provide a very low overall RMSE on the measurements.

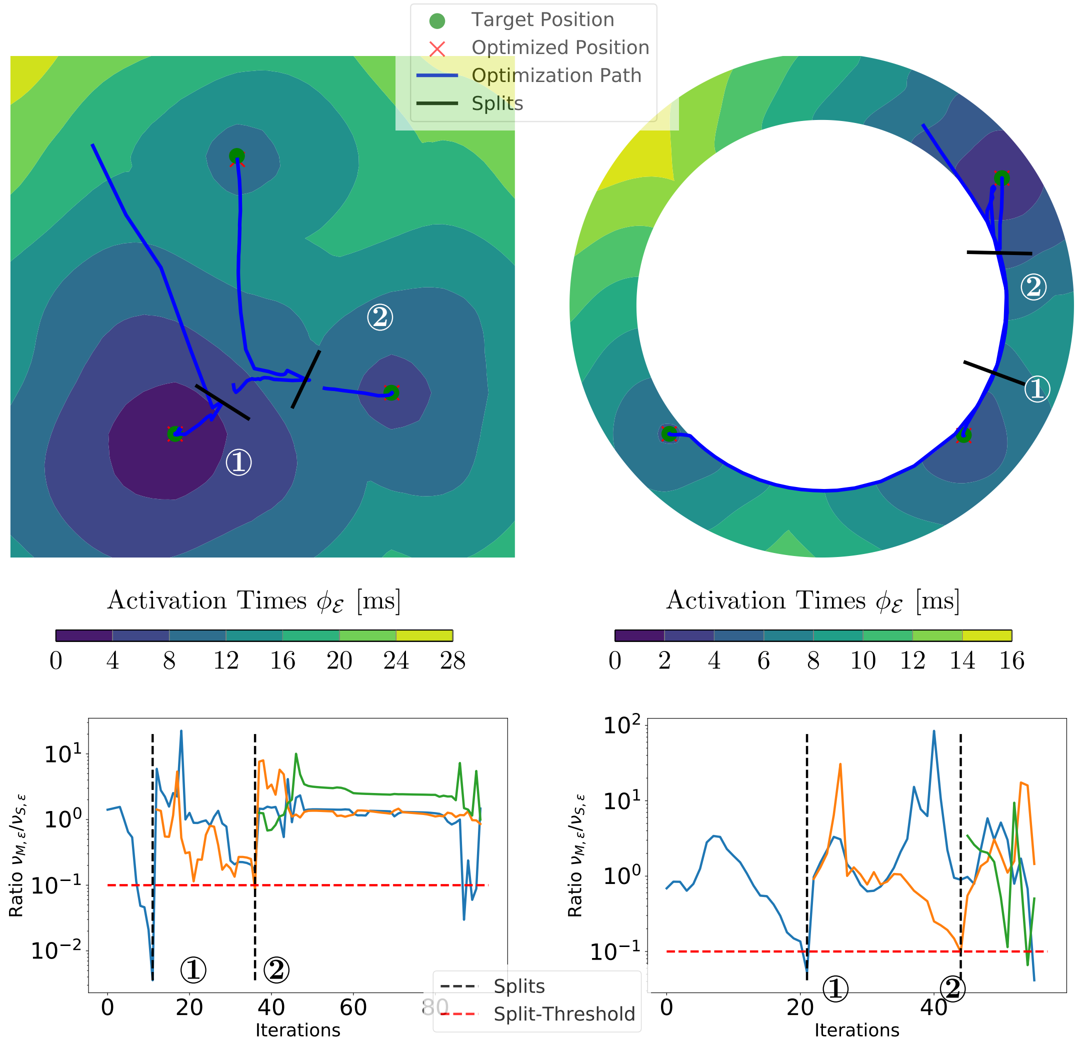


Figure 4.29: Results of the 2D experiments for the topological gradient. Top row: optimization paths starting with a single EAS. Bottom row: plots of $\frac{\nu_{M,\varepsilon}}{\nu_{S,\varepsilon}}$ for each EAS depending on the iterations, where an EAS is split if this ratio is below the dotted red line. The location (top) and iteration (bottom) of the splits are marked by ① and ②. Note that an EAS only splits if all parameters have converged (see Section 4.7.4.2).

4.7.4.3 ECG

In what follows, we present numerical results for the ECG optimization for both 2D experiments as well as the trifascicular model in a simplified fashion as a proof-of-concept. The ECG requires an additional full torso domain Ω_T and the computation of the lead fields. For all experiments in this paper, we submerge all three in-silico experiments (i.e. $\boxed{\mathbf{1}}$,$\boxed{\mathbf{2}}$,$\boxed{\mathbf{3}}$) into a non-equilateral cube-torso without any additional organs and an overall torso conductivity of 0.2 $\frac{\mathrm{S}}{\mathrm{m}}$. The size of the cube-torso is proportional to the

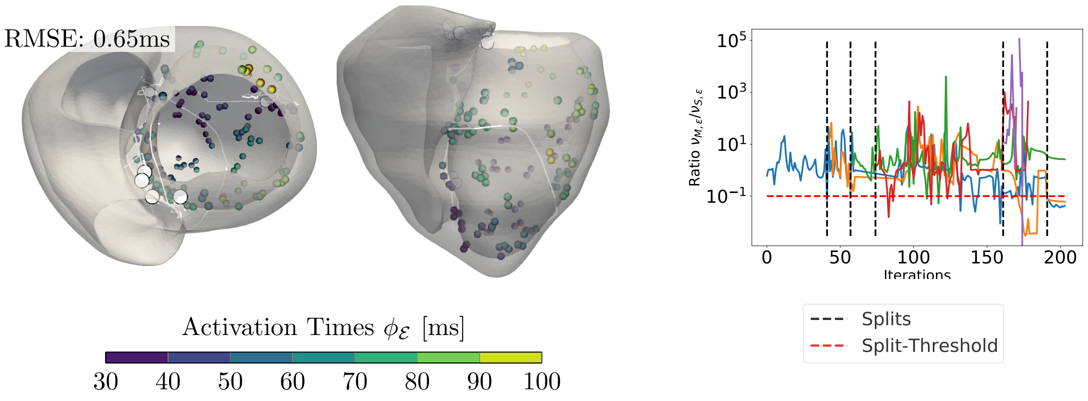


Figure 4.30: Results for the topological gradient extension on the CRT experiment with a visualization analogous to Figure 4.29.

bounding-box of Ω. The computed lead fields are shown in Figure 4.31. In all cases, we generated a noiseless target ECG from the reference model setup with parameters and initiation sites already presented in Section 4.7.4. We optimize our model with random initialization with respect to this target ECG. Note that we do not focus on the generated ECGs' absolute potentials, since this heavily depends on the actual torso setup. Instead, we rather focus on the overall morphology of the ECGs.

To compute the lead fields in (3.19), our cube-torso is sampled using a structured regular grid of 100^d equidistant points, and the problem is solved with a finite difference scheme. The lead fields are computed prior to the optimization since they remain constant. The ECG signals for the 3D models are mean-filtered with a small kernel of size ≈ 2 ms to improve accuracy.

Optimization solely based on the ECG is frequently very challenging. However, with a proper initialization $(\mathbf{x}_i, t_i)$, good fits for the ECGs can be computed. Figure 4.32 shows the optimization paths, as well as initial, target and optimized ECGs using the modified GEASI algorithm presented in Section 4.7.2.2 for the 2D examples, which are computed in approximately 2.5 hours each. The two potentials are a result of the two axis-aligned lead-fields (see Figure 4.31).

In the numerical experiments, it turned out that that the overall step size β_s has to be chosen smaller compared to the activation timing problem. The morphology of the initial ECG and the optimized ECG differ by a large margin, making the fitting non-trivial. As a result, in both the square domain and the idealized LV experiment we are able to closely match the actual sites from which the target ECG was generated (Figure 4.32, second row).

The trifascicular model in Figure 4.33 is computationally demanding since in each iteration step a computation of all geodesics is required, i.e. we need to solve $\approx 10^5$ ODEs per iteration (for further details we refer to Section 4.7.5). As each initial EAS is randomly

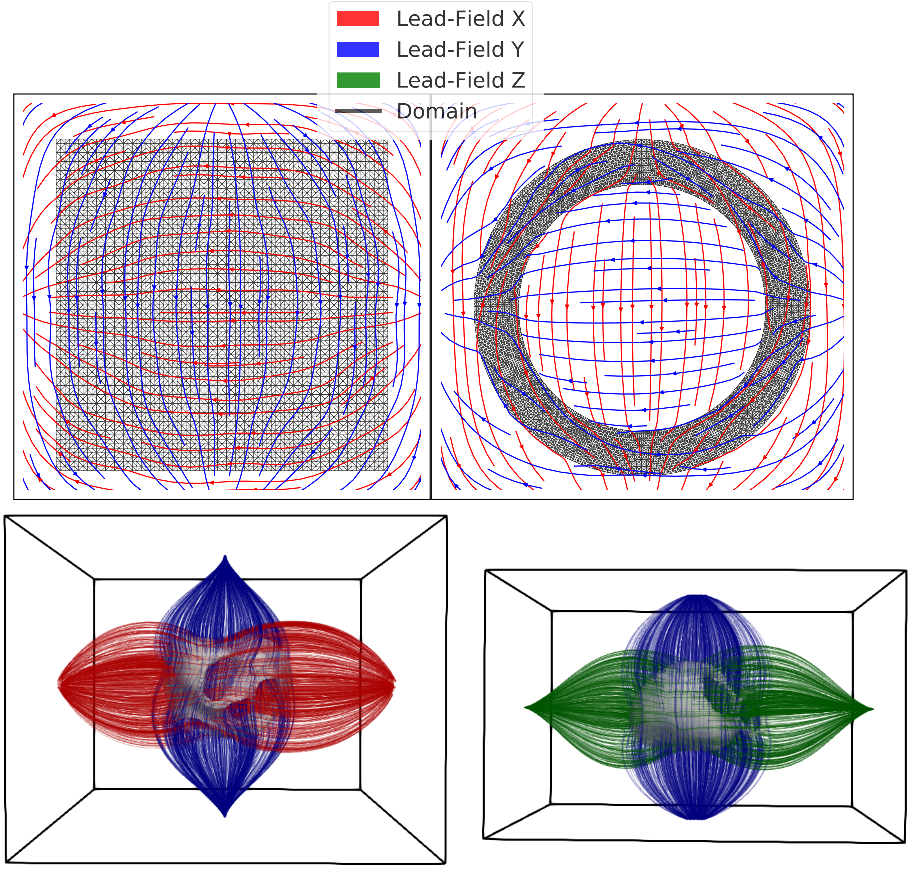


Figure 4.31: Setup for the ECG experiments showing the torso domain Ω_T. The heart domain Ω is indicated by black lines for the 2D experiments and gray silhouettes for 3D. The streamlines visualize the lead fields. Note that the lead field for axis Z (green) is only present in the 3D experiments.

chosen, the initial ECG significantly differs from the target. Note that the 3D cube torso exhibits three axis-aligned leads. Since lead-X and lead-Z have the most prominent peaks, they have the largest effect on the resulting L^2-error. After the optimization, these peaks were fitted by the algorithm by shifting most of the initiation sites to the LV and one to the anterior wall and septal region. The added difficulty with an activation featuring that many EASs is also apparent from the computed paths (white lines) which strongly vary during optimization. After termination, 4 of 6 sites are close to the ground truth sites defining the target ECG.

4.7.5 Runtime

The majority of the computational time is spent for solving the geodesics in (4.37), performed in parallel on the GPU. We highlight that the number of geodesics is proportional

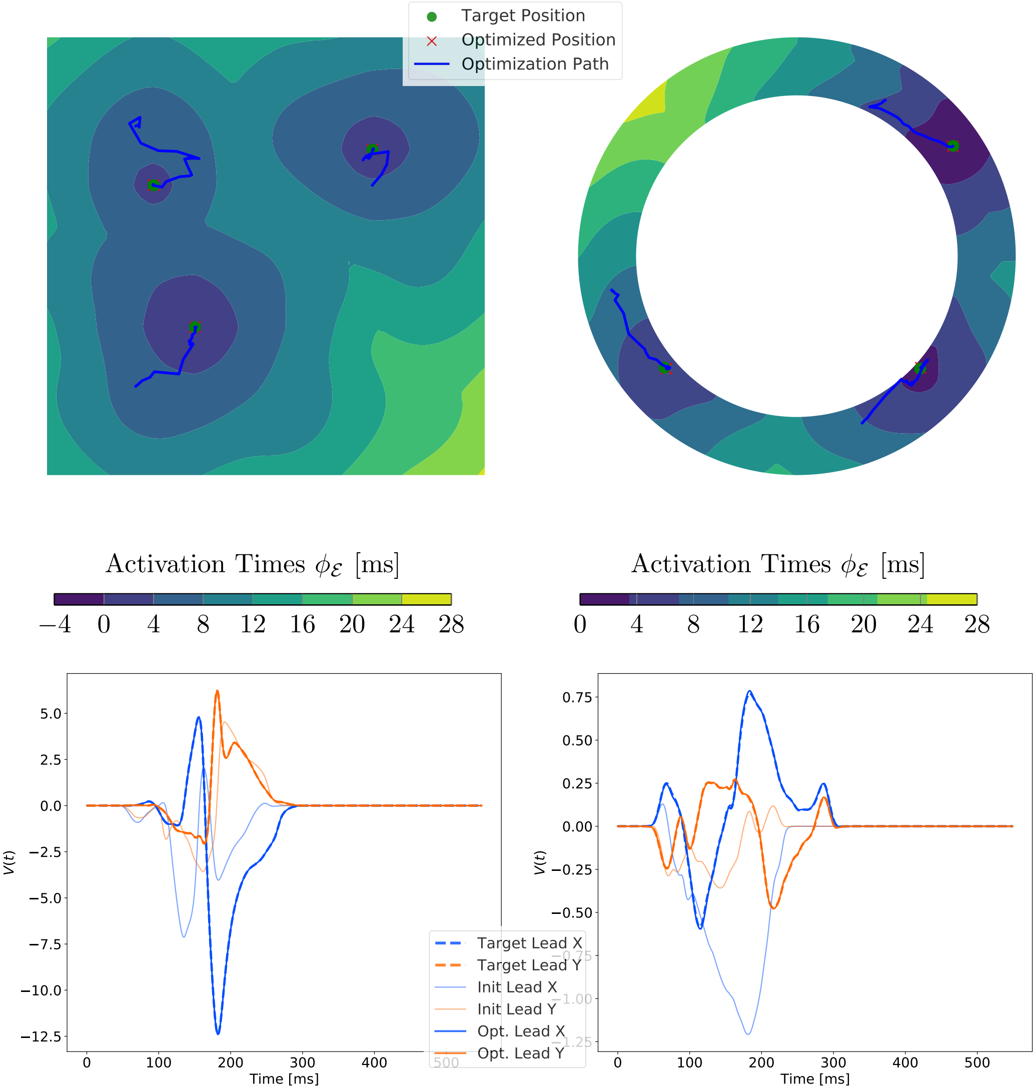


Figure 4.32: Results of the 2D ECG optimization. Top row: temporal change of the positions of the EASs along with the ground truth. Bottom row: initial, final and target ECG for fitting.

to the size of Γ in the original version (Algorithm 4.4) and proportional to Ω in the modified version (Section 4.7.2.2). The computation of all geodesics in both cases is performed in parallel on a GPU and therefore scales well with the mesh size. The bulk of computational time inside the ODE solver is spent on the projection of each ODE solution back onto the mesh and nearest neighbor computation. For the nearest neighbor computation, we implemented a custom KD-Tree implementation (publicly available on GitHub[4]). For

[4] `https://github.com/thomgrand/tf_kdtree`

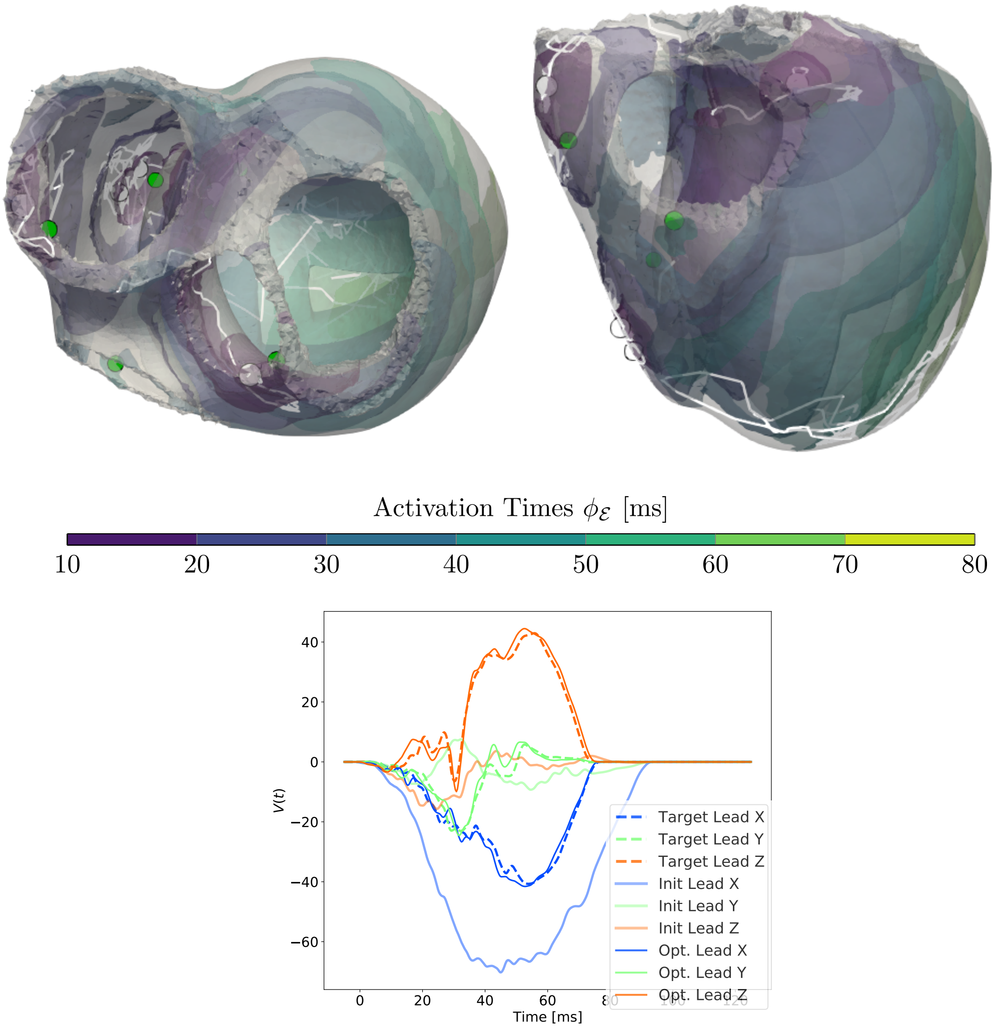


Figure 4.33: Results of the ECG optimization on the trifascicular model. Top row: optimized positions of EASs (white circles) along with temporal changes over the iterations (white lines). The green circles represent the target position from which the ground truth was generated. Bottom row: initial, final and target ECG.

the projection operator, we extract the surface of the mesh, prior to the computation using the truncated signed distance function from VTK[5]. The K-nearest neighbor elements of the current positions of the geodesics are then queried to calculate the analytical projection onto all reference elements. The projection to all nearest neighbors is the minimum distance projection onto the mesh Ω.

Solving the eikonal equation in (4.29) as well as the Gauss–Newton optimization in (4.39) only requires a minor portion of the computational time. As already mentioned, the

[5] `https://vtk.org/`

activation time optimization is much faster compared to the remaining computations. In total, the experiments were finished within about 100 iterations only taking approximately 30 minutes and 90 minutes for 2D and 3D experiments, respectively. The experiments for the topological gradient behaves similarly regarding computational time. In contrast, the 3D optimization in the ECG problem requires approximately 12 hours.

To further decrease runtime, several approaches are possible: A custom GPU implementation to solve (4.37) along with the projection could significantly speed-up the optimization. Additionally, we often witnessed a collapse of many geodesic paths, especially in the trifascicular model, making subsequent computations redundant. An adaptive sampling from the measurement domain Γ combined with a proper upsampling technique could increase performance at the cost of precision.

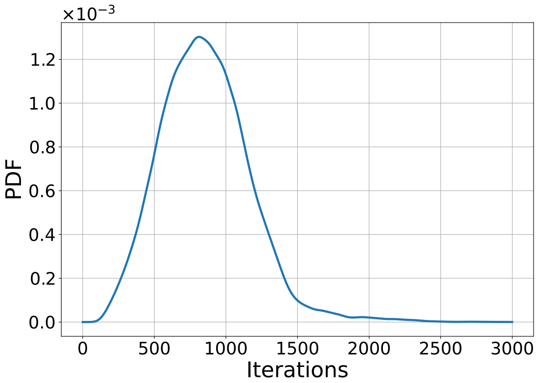


Figure 4.34: Convergence of the geodesic ODE in (4.37) for the trifascicular model over the iterations with a single EAS in the septum. The majority of the geodesics converge before 2000 iterations.

To improve performance for the 3D ECG optimization, we analyzed the convergence of the ODEs. Figure 4.34 shows a probability density function (PDF) of convergence of the geodesics γ over the number of required iterations using the trifascicular model with a single initiation site in the septum. Convergence in this case is defined as the first time two subsequent ODE iterations of (4.37) have a change of less than 10^{-10}, i.e. $||\gamma(t_{k+1}) - \gamma(t_k)|| < 10^{-10}$. We see that many of the computed geodesics converge very quickly, while points with a high geodesic distance need significantly more iterations before convergence. Our vectorized/parallel implementation to solve (4.37) exploits this fact to only include non-converged geodesics.

5 Discussion & Conclusion

> *I would rather have questions that can't be answered, than answers that can't be questioned.* (Richard P. Feynman)

In this thesis, we set out to tackle a few of the many challenges in inverse cardiac electrophysiology.

As Chapter 4 showcased, we utilized many of the theories introduced in Chapters 2 and 3 and applied them to model the problem and subsequently pose the inverse problem of identifying cardiac conduction velocities or EASs from a variety of measurements. From a modeling point of view, there are many similarities between the approaches: We assume the eikonal model as an underlying model of electrical propagation, usually solved using the FIM. An optimization problem was then formulated, usually using some form of quadratic l_2 loss on known data/measurements and subsequently solved using a first order gradient method. Locally varying parameter sets were additionally often regularized using TV, or a smooth variant thereof.

Still, the different presented methods vary greatly in application (digital twinning vs. clinical aid), abstraction level (LAT vs. ECG), dimensionality of the problem (manifolds vs. 3D meshes), optimized parameters (conduction velocity tensor vs. EAS), choice of inverse model (direct, ANN, Hamilton–jacobi formalism, ...) and used optimization approaches (FISTA, gradient descent, Gauss Newton, ...). A lot of this goes to show that the problem can be viewed from various angles that need to be handled differently. An important step in this work was to identify the key parameter set necessary for electrical activation in the heart and define it in a way that allowed for an efficient optimization given the respective data. The identified main set of parameters comes from the anisotropic eikonal equation in its most general formulation: The conduction velocity tensor and the Dirichlet boundary conditions, often referred to as EAS. Both of these parameter sets can be efficiently optimized if a LAT map is known, but the extension of optimizing against an actual ECG (see Section 4.7.2.2) remains difficult. The chosen model to compute ECGs from eikonal activation maps was a compromise between high-fidelity and computational efficiency [95]. A major challenge is the fact that the timeseries signal of an ECG is by nature very sparse in information, considering the signal remains at the baseline for a major portion of a heartbeat, whereas the major information of ventricular activation is contained in the very short QRS-complex. The employed l_2-norm compares the timeseries point-wise and is thus not able to efficiently model e.g. simple shifts of the depolarization. Additionally, second order derivatives are required for gradient based optimization of the loss functional. Further research in this field should focus on the choice of a proper, yet efficient loss functional to compare two ECGs.

As already outlined in Chapter 1, there is a plethora of applications of such a technology, such as automated digital twinning of a patient's heart. When building such models from physical assumptions that have been shown to represent (or at least approximate) encountered phenomena, we are always guaranteed to uphold a certain degree of physical fidelity. In contrast to many recent ML-orientated blackbox approaches, this has the potential to let us gain more insights into the inner workings and underlying machinery, as well as testing hypotheses. This trend is also followed by the recent advent of PINNs, which fuse ML methods with classical PDE modeling.

Two major difficulties arise in such optimizations, as we have used throughout this thesis: Firstly, fitting the actual data to the model in most cases can be difficult, since we require a certain underlying physical model. When dealing with real measurements, they are on the one hand noisy, but, on the other hand, they possibly do not fully adhere to the chosen physical model. Secondly, in many of the presented cases the problem is actually ill-posed, meaning there are infinitely many choices of parameters that could potentially model the encountered data. We usually combat this problem by means of regularization, but most studies at the date of writing are still concerned with fitting the data and do either not use any kind of regularization at all, or simple heuristics are chosen according to the experiments. Since there is no consensus (yet), what a proper regularization for inverse problems of cardiac electrophysiology should be in this case, we relied on general regularization strategies and functions for inverse problems, i.e. TV in most cases.

In some cases, it is also not advisable to perfectly fit a given ECG as there can be a lot of variation and noise present in the signal. This raises the concern that probabilistic models would be better suited to model not just a single beat, but the beat distribution of a given patient. In theory, this could also help in better modeling arrhythmias, as some of them (e.g. premature ventricular contraction) do not alter the morphology of each beat. Additionally, such probabilistic models could also quantify uncertainties of both the parameters and the resulting ECG, giving some confidence on how much the input and output of the model can be trusted.

5.1 Eikonal Equation

In this work, we rely on the anisotropic eikonal equation, but other versions thereof are also applicable. More specifically, several eikonal frameworks to model physical and medical processes have been proposed over the last three decades, which can be derived from either the monodomain or the bidomain equation using a perturbation argument [51]. The most common equation inferred from a first order approximation of the monodomain equation is the anisotropic eikonal equation (4.29). The eikonal model originating from the bidomain model is slightly different and is based on a Finsler-type metric [2].

Second-order approximations lead to the curvature-eikonal, diffusion-eikonal and viscous-eikonal equations. In the curvature-eikonal model [77], the front velocity is corrected by the curvature of the wavefront in the metric induced by the conduction velocity tensor. In contrast, in the diffusion-eikonal equation [32] a diffusion term is added to the right-hand side of (4.29). Finally, in the viscous-eikonal model [80], a squared eikonal equation is considered, which is corrected by a diffusion term.

Higher-order approximations have also been proposed, but are rarely used in practice [42]. The effect of higher-order terms is more pronounced in front collisions, at the boundary of the domain and in narrow channels, e.g. scarred tissue. In practice, however, deviations from the standard eikonal model are minimal and the standard model is therefore widely accepted for personalization of cardiac models. This goes to show that the eikonal model is a valid choice for inverse problems in cardiac electrophysiology.

5.2 ECG

The ECG results of GEASI in Section 4.7 demonstrated that inverse eikonal problems can be used to fit a model to a given ECG. However, one main problem that we faced is getting stuck in local minima. While the l_2-error is relatively low in these minima, the morphology of the optimized and the target ECG differ a lot. One of the main reasons for this problem could result from the usage of the l_2-error, which is not robust to transformations of the timeseries, such as time shifts [127]. Better error measures for this type of optimization include dynamic time warping [110] and the Wasserstein distance [22, 123]. Finally, different optimization algorithms that are less susceptible to local minima, such as the ones used in many ML applications (e.g. ADAM [79]), could further help to overcome this issue.

While computational capabilities have reached new heights for cardiac modeling, as was proven by the advent of real time electrophysiological solvers in the past years [87, 95], the inverse electrophysiological problem remains challenging. With the initial outset at the start of the ILearnHeart project, the target was to model a specific heartbeat of a patient. We achieved this goal in different levels of accuracy in the presented and published works. In order to advance to a fully automatic parameterization procedure (or digital twinning), further research is needed both in the area of constructing robust measures to compare ECGs, as well as an improved formulation of the inverse problem. This holds especially true for regularization strategies based on prior knowledge about contributing physiology (that is so far only available in few studies) and needs proper mathematical models. From the results of this thesis, I conclude and predict that the automated digital twinning for cardiac electrophysiology is within our grasp and only misses few, yet essential parts that can further help in combating the ill-posedness of the problem.

Bibliography

[1] R. A. Adams and J. J. Fournier. *Sobolev spaces*. Elsevier, 2003. (page 16)

[2] L. Ambrosio, P. Colli Franzone, and G. Savaré. On the asymptotic behaviour of anisotropic energies arising in the cardiac bidomain model. *Interfaces Free Bound.*, 2(3):213–266, 2000, `https://doi.org/10.4171/IFB/19`. (page 138)

[3] H. J. Arevalo, F. Vadakkumpadan, E. Guallar, A. Jebb, P. Malamas, K. C. Wu, and N. A. Trayanova. Arrhythmia risk stratification of patients after myocardial infarction using personalized heart models. *Nature communications*, 7:11437, May 2016. (page 4, 91)

[4] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, January 1966, `https://msp.org/pjm/1966/16-1/p01.xhtml`. (page 74)

[5] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J Matrix Anal Appl*, 29(1):328–347, 2007. (page 93)

[6] C. M. Augustin, A. Neic, M. Liebmann, A. J. Prassl, S. A. Niederer, G. Haase, and G. Plank. Anatomically accurate high resolution modeling of human whole heart electromechanics: A strongly scalable algebraic multigrid solver method for nonlinear deformation. *Journal of computational physics*, 305:622–646, January 2016. (page 76)

[7] F. Barber, I. García-Fernández, M. Lozano, and R. Sebastian. Automatic estimation of Purkinje-myocardial junction hot-spots from noisy endocardial samples: a simulation study. *Int. J. Numer. Methods Biomed. Eng.*, 34(7):e2988, 15, 2018, `https://doi.org/10.1002/cnm.2988`. (page 61)

[8] A. Barone, A. Gizzi, F. Fenton, S. Filippi, and A. Veneziani. Experimental validation of a variational data assimilation procedure for estimating space-dependent cardiac conductivities. *Comput Method Appl M*, 358:112615, 2020. (page 62, 64)

[9] J. D. Bayer, R. C. Blake, G. Plank, and N. A. Trayanova. A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. *Annals of Biomedical Engineering*, 40(10):2243–2254, October 2012. (page 61, 64, 76, 87, 125)

[10] J. Bayer, A. J. Prassl, A. Pashaei, J. F. Gomez, A. Frontera, A. Neic, G. Plank, and E. J. Vigmond. Universal ventricular coordinates: A generic framework for describing position within the heart and transferring data. *Medical image analysis*, 45:83–93, April 2018. (page 77)

[11] L. R. Bear, P. R. Huntjens, R. D. Walton, O. Bernus, R. Coronel, and R. Dubois. Cardiac electrical dyssynchrony is accurately detected by noninvasive electrocardiographic imaging. *Heart rhythm*, 15(7):1058–1069, 2018. (page 89)

[12] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009, `http://epubs.siam.org/doi/abs/10.1137/080716542`. (page 37, 40, 43, 44, 95, 96)

[13] A. Beck. *First-order methods in optimization*. SIAM, 2017. (page 5, 11, 35, 36, 37, 38, 39, 40, 45)

[14] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015. (page 5)

[15] M. J. Bishop and G. Plank. Simulating photon scattering effects in structurally detailed ventricular models using a monte carlo approach. *Frontiers in physiology*, 5:338, 2014. (page 88)

[16] F. Bornemann and C. Rasch. Finite-element discretization of static Hamilton-Jacobi equations based on a local variational principle. *Comput. Vis. Sci.*, 9(2):57–69, 2006, `https://doi.org/10.1007/s00791-006-0016-y`. (page 109)

[17] P. M. Boyle, M. Deo, G. Plank, and E. J. Vigmond. Purkinje-mediated effects in the response of quiescent ventricles to defibrillation shocks. *Annals of biomedical engineering*, 38:456–468, February 2010. (page 77)

[18] S. Brenner and R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2007. (page 29, 30)

[19] A. Bressan. Viscosity Solutions of Hamilton-Jacobi Equations and Optimal Control Problems. January 2003. (page 15, 25)

[20] J. C. Butcher and N. Goodwin. *Numerical methods for ordinary differential equations*, volume 2. Wiley Online Library, 2008. (page 18)

[21] R. H. Byrd, P. Lu, J. Nocedal, and C. Y. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995, `https://doi.org/10.1137/0916069`. (page 46, 74, 105, 124)

[22] J. Camps, B. Lawson, C. Drovandi, A. Minchole, Z. J. Wang, V. Grau, K. Burrage, and B. Rodriguez. Inference of ventricular activation properties from non-invasive electrocardiography. *arXiv:2010.15214 [eess, q-bio]*, October 2020, `http://arxiv.org/abs/2010.15214`. arXiv: 2010.15214. (page 139)

[23] C. Cantwell, C. Roney, F. Ng, J. Siggers, S. Sherwin, and N. Peters. Techniques for automated local activation time annotation and conduction velocity estimation in cardiac mapping. *Comput Biol Med*, 65:229–242, 2015. (page 96, 97)

[24] M. P. d. Carmo. *Riemannian geometry.* Birkhäuser, 1992. (page 18, 19, 20, 21)

[25] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J Math Imaging Vis*, 40(1):120–145, 2011. (page 45, 96)

[26] A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numer.*, 25:161–319, 2016, https://doi.org/10.1017/S096249291600009X. (page 46, 75, 115, 124)

[27] S. Chen, J. Montgomery, and A. Bolufé-Röhler. Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence*, 42(3):514–526, April 2015, https://doi.org/10.1007/s10489-014-0613-2. (page 5)

[28] C. Chicone. *Ordinary differential equations with applications*, volume 34. Springer Science & Business Media, 2006. (page 16)

[29] L. Clerc. Directional differences of impulse spread in trabecular muscle from mammalian heart. *J. Physiol*, 255(2):335–346, 1976. (page 3)

[30] M. Cluitmans, D. H. Brooks, R. MacLeod, O. Doessel, M. S. Guillem, P. M. van Dam, J. Svehlikova, B. He, J. Sapp, L. Wang, and L. Bear. Validation and opportunities of electrocardiographic imaging: From technical achievements to clinical applications. *Frontiers in physiology*, 9:1305, 2018. (page 91)

[31] E. A. Coddington and N. Levinson. *Theory of ordinary differential equations.* Tata McGraw-Hill Education, 1955. (page 17)

[32] P. Colli Franzone, L. Guerri, and S. Rovida. Wavefront propagation in an activation model of the anisotropic cardiac tissue: asymptotic analysis and numerical simulations. *Journal of mathematical biology*, 28(2):121–176, 1990. (page 3, 61, 107, 139)

[33] P. Colli Franzone and L. Guerri. Spreading of excitation in 3-d models of the anisotropic cardiac tissue. I. validation of the eikonal model. *Mathematical Biosciences*, 113(2):145–209, February 1993, http://www.sciencedirect.com/science/article/pii/002555649390001Q. (page 4, 54, 63, 64, 76, 107)

[34] J. B. Conway. *A course in functional analysis*, volume 96. Springer, 2019. (page 10)

[35] J. Corral-Acero, F. Margara, M. Marciniak, C. Rodero, F. Loncaric, Y. Feng, A. Gilbert, J. F. Fernandes, H. A. Bukhari, A. Wajdan, M. V. Martinez, M. S. Santos, M. Shamohammdi, H. Luo, P. Westphal, P. Leeson, P. DiAchille, V. Gurev, M. Mayr, L. Geris, P. Pathmanathan, T. Morrison, R. Cornelussen, F. Prinzen, T. Delhaas, A. Doltra, M. Sitges, E. J. Vigmond, E. Zacur, V. Grau, B. Rodriguez, E. W. Remme, S. Niederer, P. Mortier, K. McLeod, M. Potse, E. Pueyo, A. Bueno-Orovio, and P. Lamata. The 'Digital Twin' to enable the vision of precision cardiology. *European Heart Journal*, 41(48):4556–4564, 03 2020, `https://doi.org/10.1093/eurheartj/ehaa159`. (page 3)

[36] R. Courant and E. J. McShane. *Differential and integral calculus*, volume 1. Wiley Online Library, 1937. (page 13)

[37] M. G. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 282(2):487–502, 1984, `https://doi.org/10.2307/1999247`. (page 107, 108)

[38] M. H. Crawford and M.-H. Education. *Current diagnosis & treatment cardiology*. Lange Medical Books/McGraw-Hill, 2003. (page 3)

[39] A. Crozier, C. M. Augustin, A. Neic, A. J. Prassl, M. Holler, T. E. Fastl, A. Hennemuth, K. Bredies, T. Kuehne, M. J. Bishop, S. A. Niederer, and G. Plank. Image-based personalization of cardiac anatomy for coupled electromechanical modeling. *Annals of biomedical engineering*, 44:58–70, January 2016. (page 76)

[40] J. C. Demoulin and H. E. Kulbertus. Histopathological examination of concept of left hemiblock. *British heart journal*, 34:807–814, August 1972. (page 76)

[41] J. Dhamala, P. Bajracharya, H. J. Arevalo, J. Sapp, B. M. Horácek, K. C. Wu, N. A. Trayanova, and L. Wang. Embedding high-dimensional Bayesian optimization via generative modeling: Parameter personalization of cardiac electrophysiological models. *Medical Image Analysis*, 62:101670, May 2020, `https://www.sciencedirect.com/science/article/pii/S1361841520300360`. (page 62)

[42] H. Dierckx, O. Bernus, and H. Verschelde. Accurate eikonal-curvature relation for wave fronts in locally anisotropic reaction-diffusion systems. *Physical review letters*, 107(10):108101, 2011. (page 139)

[43] I. C. Dolcetta and P. L. Lions. *Viscosity solutions and applications: lectures given at the 2nd Session of the Centro internazionale matematico estivo (CIME) held at Montecatini Terme, Italy, June 12-20, 1995*. Springer, 1997. (page 14, 25)

[44] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, 1995, `https://doi.org/10.1109/18.382009`. (page 124)

[45] M. H. Draper and S. Weidmann. Cardiac resting and action potentials recorded with an intracellular electrode. *The Journal of physiology*, 115:74–94, September 1951. (page 77)

[46] W. J. Duffin. *Electricity and magnetism.* McGraw-Hill, 1980. (page 57)

[47] D. Durrer, R. T. van Dam, G. E. Freud, M. J. Janse, F. L. Meijler, and R. C. Arzbaecher. Total excitation of the isolated human heart. *Circulation*, 41:899–912, June 1970. (page 76, 77, 87)

[48] B. Erem, P. M. van Dam, and D. H. Brooks. Identifying model inaccuracies and solution uncertainties in noninvasive activation-based imaging of cardiac excitation using convex relaxation. *IEEE transactions on medical imaging*, 33:902–912, April 2014. (page 62, 89)

[49] L. C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010, `https://doi.org/10.1090/gsm/019`. (page 9, 10, 23, 24, 26, 29)

[50] B. Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, 1988, `https://www.ams.org/mcom/1988-51-184/S0025-5718-1988-0935077-0/`. (page 28)

[51] P. C. Franzone, L. F. Pavarino, and S. Scacchi. *Mathematical cardiac electrophysiology*, volume 13. Springer, 2014. (page 3, 47, 48, 50, 55, 56, 57, 118, 120, 138)

[52] Z. Fu, W. Jeong, Y. Pan, R. Kirby, and R. Whitaker. A Fast Iterative Method for Solving the Eikonal Equation on Triangulated Surfaces. *SIAM Journal on Scientific Computing*, 33(5):2468–2488, January 2011, `https://epubs.siam.org/doi/abs/10.1137/100788951`. (page 63, 65, 67, 92, 94, 156, 159)

[53] Z. Fu, R. Kirby, and R. Whitaker. A Fast Iterative Method for Solving the Eikonal Equation on Tetrahedral Domains. *SIAM Journal on Scientific Computing*, 35(5):C473–C494, January 2013, `https://epubs.siam.org/doi/abs/10.1137/120881956`. (page 54, 63, 64, 65, 67, 68, 70, 159)

[54] D. B. Geselowitz. Description of cardiac sources in anisotropic cardiac muscle. application of bidomain model. *Journal of electrocardiology*, 25 Suppl:65–67, 1992. (page 89)

[55] A. Gharaviri, E. Bidar, M. Potse, S. Zeemering, S. Verheule, S. Pezzuto, R. Krause, J. G. Maessen, A. Auricchio, and U. Schotten. Epicardial fibrosis explains increased endo–epicardial dissociation and epicardial breakthroughs in human atrial fibrillation. *Front Phys*, 11:68, 2020. (page 96)

[56] S. Giffard-Roisin, H. Delingette, T. Jackson, J. Webb, L. Fovargue, J. Lee, C. A. Rinaldi, R. Razavi, N. Ayache, and M. Sermesant. Transfer learning from simulations on a reference anatomy for ECGI in personalized cardiac resynchronization therapy. *IEEE Transactions on Biomedical Engineering*, 66(2):343–353, 2019. (page 64)

[57] D. Gilbarg and N. S. Trudinger. *Elliptic partial differential equations of second order*. Classics in Mathematics. Springer-Verlag, Berlin, 2001. Reprint of the 1998 edition. (page 118, 119)

[58] T. Grandits, A. Effland, T. Pock, R. Krause, G. Plank, and S. Pezzuto. GEASI: Geodesic-based Earliest Activation Sites Identification in cardiac models. *International Journal for Numerical Methods in Biomedical Engineering (accepted for publication)*, February 2021, `http://arxiv.org/abs/2102.09962`. arXiv: 2102.09962. (page 8, 106)

[59] T. Grandits, K. Gillette, A. Neic, J. Bayer, E. Vigmond, T. Pock, and G. Plank. An inverse Eikonal method for identifying ventricular activation sequences from epicardial activation maps. *Journal of Computational Physics*, 419:109700, October 2020, `http://www.sciencedirect.com/science/article/pii/S0021999120304745`. (page 7, 45, 64, 67, 70)

[60] T. Grandits, S. Pezzuto, F. S. Costabal, P. Perdikaris, T. Pock, G. Plank, and R. Krause. Learning Atrial Fiber Orientations and Conductivity Tensors from Intracardiac Maps Using Physics-Informed Neural Networks. In D. B. Ennis, L. E. Perotti, and V. Y. Wang, editors, *Functional Imaging and Modeling of the Heart*, Lecture Notes in Computer Science, pages 650–658, Cham, 2021. Springer International Publishing. (page 8, 17, 63, 91, 102)

[61] T. Grandits, S. Pezzuto, J. M. Lubrecht, T. Pock, G. Plank, and R. Krause. PIEMAP: Personalized Inverse Eikonal Model from Cardiac Electro-Anatomical Maps. In E. Puyol Anton, M. Pop, M. Sermesant, V. Campello, A. Lalande, K. Lekadir, A. Suinesiaputra, O. Camara, and A. Young, editors, *Statistical Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges*, Lecture Notes in Computer Science, pages 76–86, Cham, 2021. Springer International Publishing. (page 7, 8, 45, 63, 64, 91, 93, 105, 106)

[62] T. Grandits and T. Pock. Optimizing Wavelet Bases for Sparser Representations. In M. Pelillo and E. Hancock, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science, pages 249–262, Cham, 2018. Springer International Publishing. (page)

[63] Guillem Maria S., Climent Andreu M., Millet Jose, Arenal Ángel, Fernández-Avilés Francisco, Jalife José, Atienza Felipe, and Berenfeld Omer. Noninvasive Localization of Maximal Frequency Sites of Atrial Fibrillation by Body Surface Potential

Mapping. *Circulation: Arrhythmia and Electrophysiology*, 6(2):294–301, April 2013, `https://www.ahajournals.org/doi/full/10.1161/circep.112.000167`. Publisher: American Heart Association. (page 62)

[64] J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, pages 49–52, 1902. (page 41)

[65] M. Haissaguerre, E. Vigmond, B. Stuyvers, M. Hocini, and O. Bernus. Ventricular arrhythmias and the his-purkinje system. *Nature reviews. Cardiology*, 13:155–166, March 2016. (page 76)

[66] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018. Publisher: Wiley Online Library. (page 63)

[67] C. Han, S. M. Pogwizd, L. Yu, Z. Zhou, C. R. Killingsworth, and B. He. Imaging cardiac activation sequence during ventricular tachycardia in a canine model of nonischemic heart failure. *American journal of physiology. Heart and circulatory physiology*, 308:H108–H114, January 2015. (page 62, 89)

[68] B. He, G. Li, and X. Zhang. Noninvasive imaging of cardiac transmembrane potentials within three-dimensional myocardium by means of a realistic geometry anisotropic heart model. *IEEE transactions on bio-medical engineering*, 50:1190–1202, October 2003. (page 62, 89)

[69] M. R. Hestenes, E. Stiefel, et al. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952. (page 35)

[70] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, August 1952, `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/`. (page 47, 51)

[71] J. Jalife, M. Delmar, J. Anumonwo, O. Berenfeld, and J. Kalifa. *Basic cardiac electrophysiology for the clinician*. John Wiley & Sons, 2011. (page 48, 49, 61)

[72] A. M. Janssen, D. Potyagaylo, O. Dössel, and T. F. Oostendorp. Assessment of the equivalent dipole layer source model in the reconstruction of cardiac activation times on the basis of bspms produced by an anisotropic model of the heart. *Medical & biological engineering & computing*, 56:1013–1025, June 2018. (page 62, 89)

[73] W. Jeong and R. Whitaker. A Fast Iterative Method for Eikonal Equations. *SIAM Journal on Scientific Computing*, 30(5):2512–2534, January 2008, `https://epubs.siam.org/doi/abs/10.1137/060670298`. (page 63, 65)

[74] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. Publisher: American Association for the Advancement of Science. (page 63)

[75] S. Kallhovd, M. M. Maleckar, and M. E. Rognes. Inverse estimation of cardiac activation times via gradient-based optimization. *International journal for numerical methods in biomedical engineering*, 34(2):e2919, 2018. (page 64)

[76] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, January 1998, `https://epubs.siam.org/doi/10.1137/S1064827595287997`. (page 78)

[77] J. P. Keener. An eikonal-curvature equation for action potential propagation in myocardium. *J. Math. Biol.*, 29(7):629–651, 1991, `https://doi.org/10.1007/BF00163916`. (page 107, 121, 139)

[78] J. P. Keener and J. Sneyd. *Mathematical physiology*, volume 1. Springer, 1998. (page 3, 47, 48, 49, 50, 51, 53, 54, 56)

[79] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017, `http://arxiv.org/abs/1412.6980`. arXiv: 1412.6980. (page 5, 63, 105, 139)

[80] K. Kunisch, A. Neic, G. Plank, and P. Trautmann. Inverse localization of earliest cardiac activation sites from activation maps based on the viscous Eikonal equation. *J. Math. Biol.*, 79(6-7):2033–2068, 2019, `https://doi.org/10.1007/s00285-019-01419-3`. (page 64, 139)

[81] W. Kühnel. *Differential geometry*, volume 77. American Mathematical Soc., 2015. (page 18)

[82] M. G. Larson and F. Bengzon. *The finite element method: theory, implementation, and applications*, volume 10 of *Texts in Computational Science and Engineering*. Springer, Heidelberg, 2013, `https://doi.org/10.1007/978-3-642-33287-6`. (page 29, 30, 122)

[83] J. M. Lubrecht, T. Grandits, A. Gharaviri, U. Schotten, T. Pock, G. Plank, R. Krause, A. Auricchio, G. Conte, and S. Pezzuto. Automatic reconstruction of the left atrium activation from sparse intracardiac contact recordings by inverse estimate of fibre structure and anisotropic conduction in a patient-specific model. *EP Europace*, 23(Supplement_1):i63–i70, March 2021, `https://doi.org/10.1093/europace/euaa392`. (page 7, 8, 99, 100, 102, 103)

[84] C.-h. Luo and Y. Rudy. A dynamic model of the cardiac ventricular action potential. I. Simulations of ionic currents and concentration changes. *Circulation research*, 74(6):1071–1096, 1994. Publisher: Am Heart Assoc. (page 51)

[85] R. McFee and F. D. Johnston. Electrocardiographic leads: I. introduction. *Circulation*, 8(4):554–568, 1953. Publisher: Am Heart Assoc. (page 59)

[86] C. C. Mitchell and D. G. Schaeffer. A two-current model for the dynamics of cardiac membrane. *Bulletin of Mathematical Biology*, 65(5):767–793, September 2003, `https://doi.org/10.1016/S0092-8240(03)00041-7`. (page 48, 49, 50, 51)

[87] A. Neic, F. O. Campos, A. J. Prassl, S. A. Niederer, M. J. Bishop, E. J. Vigmond, and G. Plank. Efficient computation of electrograms and ECGs in human whole heart simulations using a reaction-eikonal model. *Journal of Computational Physics*, 346:191–211, October 2017, `http://www.sciencedirect.com/science/article/pii/S0021999117304655`. (page 3, 89, 107, 117, 139)

[88] S. A. Niederer, J. Lumens, and N. A. Trayanova. Computational models in cardiology. *Nature Reviews Cardiology*, 16(2):100–111, February 2019, `https://www.nature.com/articles/s41569-018-0104-y`. (page 3)

[89] V. K. Nimmagadda, A. Akoglu, S. Hariri, and T. Moukabary. Cardiac simulation on multi-GPU platform. *The Journal of Supercomputing*, 59(3):1360–1378, March 2012, `https://doi.org/10.1007/s11227-010-0540-x`. (page 4)

[90] P. J. Olver. *Introduction to partial differential equations*. Springer, 2014. (page 28)

[91] B. O'neill. *Semi-Riemannian geometry with applications to relativity*. Academic press, 1983. (page 18, 20, 112)

[92] N. Ono, T. Yamaguchi, H. Ishikawa, M. Arakawa, N. Takahashi, T. Saikawa, and T. Shimada. Morphological varieties of the purkinje fiber network in mammalian hearts, as revealed by light and electron microscopy. *Archives of histology and cytology*, 72:139–149, 2009. (page 76)

[93] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. Publisher: Elsevier. (page 54)

[94] S. Palamara, C. Vergara, E. Faggiano, and F. Nobile. An effective algorithm for the generation of patient-specific Purkinje networks in computational electrocardiology. *J. Comput. Phys.*, 283:495–517, 2015, `https://doi.org/10.1016/j.jcp.2014.11.043`. (page 61, 62)

[95] S. Pezzuto, P. Kal'avský, M. Potse, F. W. Prinzen, A. Auricchio, and R. Krause. Evaluation of a Rapid Anisotropic Model for ECG Simulation. *Frontiers in Physiology*, 8:265, 2017. (page 3, 89, 107, 117, 119, 137, 139)

[96] S. Pezzuto, F. W. Prinzen, M. Potse, F. Maffessanti, F. Regoli, M. L. Caputo, G. Conte, R. Krause, and A. Auricchio. Reconstruction of three-dimensional biventricular activation based on the 12-lead electrocardiogram via patient-specific modelling, 2020. in press. (page 64)

[97] M. Potse. Scalable and accurate ecg simulation for reaction-diffusion models of the human heart. *Frontiers in physiology*, 9:370, 2018. (page 89)

[98] A. Prakosa, H. J. Arevalo, D. Deng, P. M. Boyle, P. P. Nikolov, H. Ashikaga, J. J. Blauer, E. Ghafoori, C. J. Park, R. C. Blake, F. T. Han, R. S. MacLeod, H. R. Halperin, D. J. Callans, R. C. Ranjan, S. Jonathan Nazarian, and N. A. Trayanova. Personalized virtual-heart technology for guiding the ablation of infarct-related ventricular tachycardia. *Nature Biomedical Engineering*, 2(10):732, 2018. (page 4)

[99] A. J. Prassl, F. Kickinger, H. Ahammer, V. Grau, J. E. Schneider, E. Hofer, E. J. Vigmond, N. A. Trayanova, and G. Plank. Automatically generated, anatomically accurate meshes for cardiac electrophysiology problems. *IEEE transactions on biomedical engineering*, 56:1318–1330, May 2009. (page 76)

[100] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comp. Phys.*, 378:686–707, 2019. (page 17, 63, 100, 101, 103, 104)

[101] C. A. Remme, A. O. Verkerk, W. M. H. Hoogaars, W. T. J. Aanhaanen, B. P. Scicluna, C. Annink, M. J. B. van den Hoff, A. A. M. Wilde, T. A. B. van Veen, M. W. Veldkamp, J. M. T. de Bakker, V. M. Christoffels, and C. R. Bezzina. The cardiac sodium channel displays differential distribution in the conduction system and transmural heterogeneity in the murine ventricular myocardium. *Basic research in cardiology*, 104:511–522, September 2009. (page 77)

[102] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. (page 14, 36)

[103] C. H. Roney, J. Whitaker, I. Sim, L. O'Neill, R. K. Mukherjee, O. Razeghi, E. J. Vigmond, M. Wright, M. D. O'Neill, S. E. Williams, and S. A. Niederer. A technique for measuring anisotropy in atrial conduction to estimate conduction velocity and atrial fibre direction. *Comput Biol Med*, 104:278–290, 2019. (page 62)

[104] M. B. Rosenbaum, M. V. Elizari, J. O. Lazzari, G. J. Nau, R. J. Levi, and M. S. Halpern. Intraventricular trifascicular blocks. the syndrome of right bundle branch

block with intermittent left anterior and posterior hemiblock. *American heart journal*, 78:306–317, September 1969. (page 76)

[105] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. Publisher: American Psychological Association. (page 63)

[106] G. A. Roth, C. Johnson, A. Abajobir, F. Abd-Allah, S. F. Abera, G. Abyu, M. Ahmed, B. Aksut, T. Alam, K. Alam, et al. Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015. *Journal of the American College of Cardiology*, 70(1):1–25, 2017. (page 3)

[107] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. In *Proceedings of the Eleventh Annual International Conference of the Center for Nonlinear Studies on Experimental Mathematics : Computational Issues in Nonlinear Science: Computational Issues in Nonlinear Science*, pages 259–268, New York, NY, USA, 1992. Elsevier North-Holland, Inc., `http://dl.acm.org/citation.cfm?id=142269.142312`. event-place: Los Alamos, New Mexico, USA. (page 42, 45, 73)

[108] Y. Rudy and J. E. Burnes. Noninvasive electrocardiographic imaging. *Ann Noninvasive Electrocardiol*, 4:340–358, 1999. (page 89)

[109] F. Sahli Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl. Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, 8:42, 2020. (page 17, 62, 63, 96, 97, 104, 105)

[110] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978. Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing. (page 139)

[111] M. Schwenke. Anisotropic fast marching in medical imaging applications. Master's thesis, Fraunhofer Mevis, 2010. (page 156)

[112] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci. U.S.A.*, 93(4):1591–1595, 1996, `https://doi.org/10.1073/pnas.93.4.1591`. (page 54, 63, 125)

[113] N. Sharp, Y. Soliman, and K. Crane. The vector heat method. *ACM Transactions on Graphics (TOG)*, 38(3):1–19, 2019. (page 21, 93)

[114] S. Stella, C. Vergara, M. Maines, D. Catanzariti, P. C. Africa, C. Demattè, M. Centonze, F. Nobile, M. Del Greco, and A. Quarteroni. Integration of activation maps of epicardial veins in computational cardiac electrophysiology. *Computers in Biology and Medicine*, page 104047, 2020. (page 64)

[115] D. D. Streeter, H. M. Spotnitz, D. P. Patel, J. Ross, and E. H. Sonnenblick. Fiber Orientation in the Canine Left Ventricle during Diastole and Systole. *Circ. Res.*, 24(3):339–347, 1969. (page 61, 64)

[116] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K.-A. Mardal, and A. Tveito. *Computing the electrical activity in the heart*, volume 1. Springer Science & Business Media, 2007. (page 57, 58)

[117] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. (page 43)

[118] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, September 1995. (page 54, 63, 125)

[119] J. N. Ulysses, L. A. Berg, E. M. Cherry, B. R. Liu, R. W. Dos Santos, B. G. de Barros, B. M. Rocha, and R. A. de Queiroz. An optimization-based algorithm for the construction of cardiac purkinje network models. *IEEE Transactions on Biomedical Engineering*, 65(12):2760–2768, 2018. (page 61)

[120] P. M. van Dam, T. F. Oostendorp, A. C. Linnenbank, and A. van Oosterom. Non-invasive imaging of cardiac activation and recovery. *Annals of biomedical engineering*, 37:1739–1756, September 2009. (page 62, 89)

[121] T. van Leeuwen and F. J. Herrmann. A penalty method for PDE-constrained optimization in inverse problems. *Inverse Problems*, 32(1):015007, 2015. Publisher: IOP Publishing. (page 63)

[122] G. Vigueras, I. Roy, A. Cookson, J. Lee, N. Smith, and D. Nordsletten. Toward GPGPU accelerated human electromechanical cardiac simulations. *International Journal for Numerical Methods in Biomedical Engineering*, 30(1):117–134, January 2014, `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4016759/`. (page 4)

[123] C. Villani. *Optimal transport*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2009, `https://doi.org/10.1007/978-3-540-71050-9`. Old and new. (page 139)

[124] L. Wang, H. Zhang, K. C. L. Wong, H. Liu, and P. Shi. Physiological-model-constrained noninvasive reconstruction of transmembrane potentials. *IEEE transactions on bio-medical engineering*, 57:296–315, February 2010. (page 62, 89)

[125] D. Werner. *Funktionalanalysis*. Springer, 2006. (page 11)

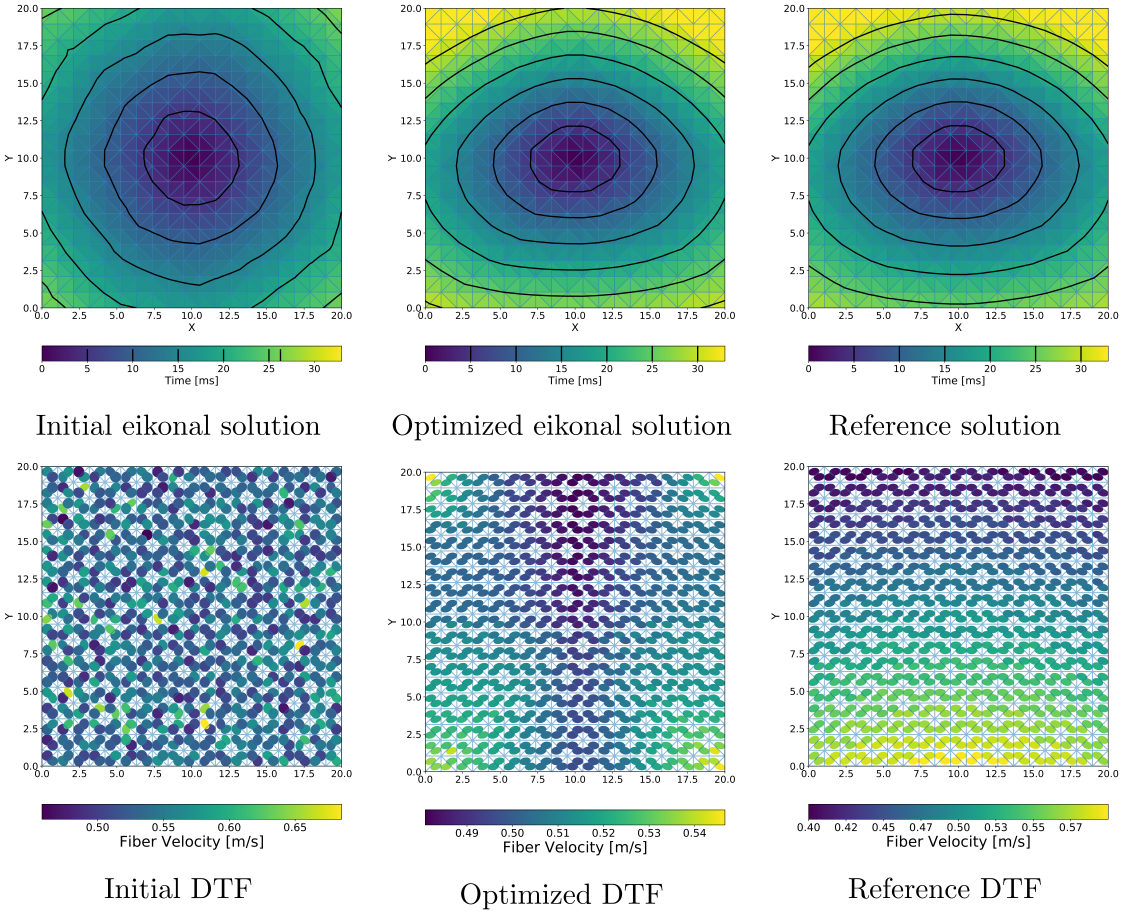
[126] H. Yang and A. Veneziani. Efficient estimation of cardiac conductivities via POD-DEIM model order reduction. *Applied Numerical Mathematics*, 115:180–199, May 2017, `https://www.sciencedirect.com/science/article/pii/S0168927417300089`. (page 62)

[127] Y. Yang, B. Engquist, J. Sun, and B. F. Hamfeldt. Application of optimal transport and the quadratic Wasserstein metric to full-waveform inversion. *GEOPHYSICS*, 83(1):R43–R62, October 2017, `https://library.seg.org/doi/abs/10.1190/geo2016-0663.1`. Publisher: Society of Exploration Geophysicists. (page 139)

[128] O. Zettinig, T. Mansi, D. Neumann, B. Georgescu, S. Rapaka, P. Seegerer, E. Kayvanpour, F. Sedaghat-Hamedani, A. Amr, J. Haas, H. Steen, H. Katus, B. Meder, N. Navab, A. Kamen, and D. Comaniciu. Data-driven estimation of cardiac electrical diffusivity from 12-lead ecg signals. *Medical image analysis*, 18:1361–1376, December 2014. (page 89)

A FIMIN

A.1 Additional 2D Experiments



Initial eikonal solution

Optimized eikonal solution

Reference solution

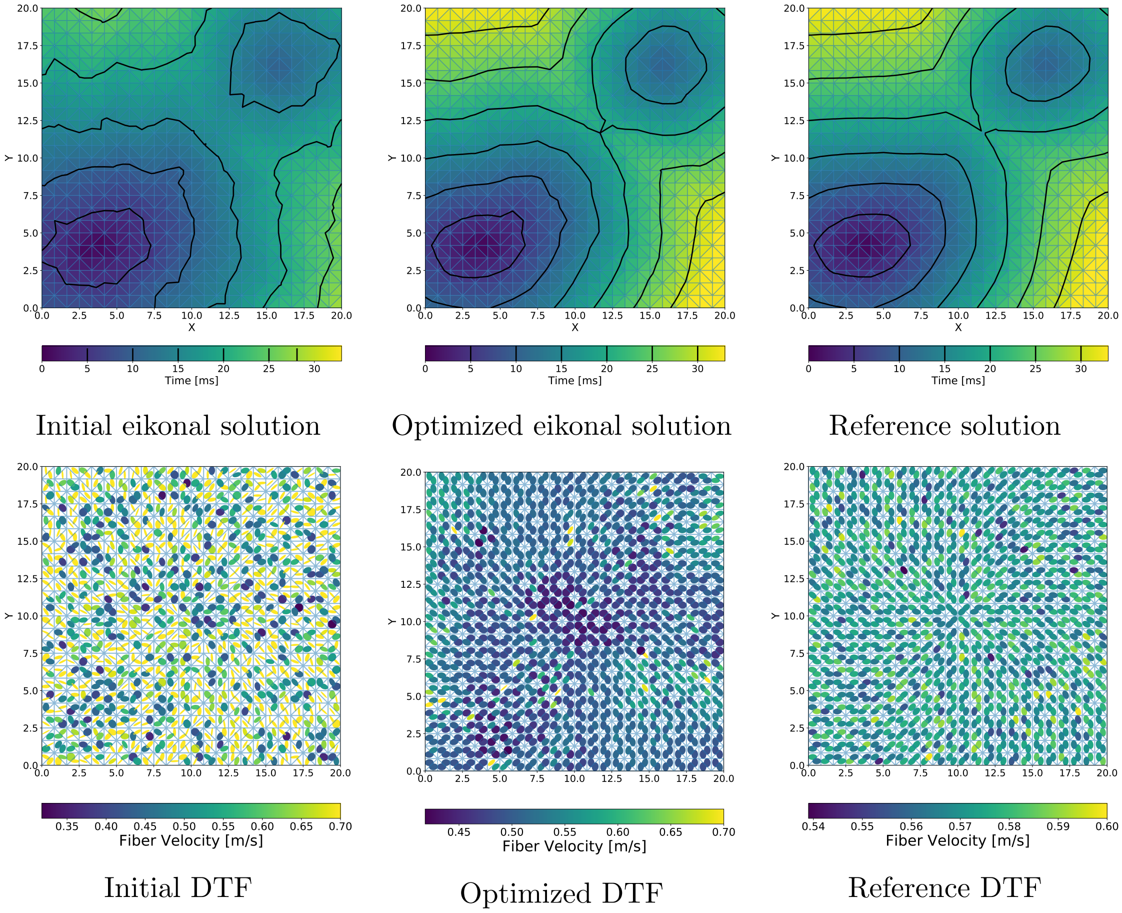
Initial DTF

Optimized DTF

Reference DTF

Figure A.11: This experiment uses curvature in horizontal direction, as well as a fiber velocity gradient along vertical direction. We are able to reconstruct the curvature of the DTF and some parts of the velocity gradients. The upper part of the domain has slightly too high fiber velocities, compensated by overall lower fiber velocity across the domain.

This section contains additional 2D experiments that demonstrate the results to be expected from the method presented in this paper. Figure A.11 is uses simple curvature of fiber, together with a velocity gradient in vertical direction, while Figure A.12 tries to reflect the joining of fibers in the heart's apex. Both examples can be reconstructed well from only boundary data, were the mismatch of activatoin times is close to zero, but there is some deviation in velocities and fiber direction, due to ill-posedness of the problem.



Initial eikonal solution

Optimized eikonal solution

Reference solution

Initial DTF

Optimized DTF

Reference DTF

Figure A.12: This experiment replicates a projection from below of the heart's apex: The fibers join in a circular fashion in the middle of the domain, fiber velocities are constant. Both fiber velocity and direction can be roughly reconstructed.

A.2 Optimal Choice of λ

This section will show how we arrive at the optimal choice of λ in Equation (4.3). The used notation is a combination of the notations in [111] and [52], on which we based our solution to the problem. We start with our optimization problem, posed in Equation (4.3):

$$\phi_4 = \min_{\lambda_1,\lambda_2} \lambda_1\phi_1 + \lambda_2\phi_2 + (1-\lambda_1-\lambda_2)\phi_3 + \sqrt{\mathbf{e}_\Delta^\top D_j \mathbf{e}_\Delta}\,, \quad \text{s.t. } \lambda_i \in [0,1] \tag{A.1}$$

$$\begin{aligned}\mathbf{e}_\Delta &= \mathbf{v}_4 - \sum_{i=1}^{3} \lambda_i \mathbf{v}_i = \underbrace{\mathbf{v}_4 - \mathbf{v}_3}_{\mathbf{w}_3} + \lambda_1 \underbrace{(\mathbf{v}_3 - \mathbf{v}_1)}_{\mathbf{w}_1} + \lambda_2 \underbrace{(\mathbf{v}_3 - \mathbf{v}_2)}_{\mathbf{w}_2} \\ &= \lambda_1\mathbf{w}_1 + \lambda_2\mathbf{w}_2 + \mathbf{w}_3\end{aligned} \tag{A.2}$$

where $\mathbf{v}_i \in \mathbb{R}^3$ are the vertices (corners) of our Tetrahedron. The wave propagation is calculated from the triangle formed by $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ to the vertex $\mathbf{v}_4$.

$$\begin{aligned}\phi_4 &= \min_{\lambda_1,\lambda_2} \lambda_1\phi_1 + \lambda_2\phi_2 + (1-\lambda_1-\lambda_2)\phi_3 + \sqrt{\mathbf{e}_\Delta^\top D_j \mathbf{e}_\Delta} \\ &= \min_{\lambda_1,\lambda_2} \lambda_1 (\phi_1 - \phi_3) + \lambda_2 (\phi_2 - \phi_3) + \phi_3 + ||\lambda_1\mathbf{w}_1 + \lambda_2\mathbf{w}_2 + \mathbf{w}_3||_{2,D}\end{aligned} \tag{A.3}$$

We state the unconstrained optimality condition as:

$$\begin{aligned}\nabla\phi_4 &= \begin{pmatrix} \phi_1 - \phi_3 + \frac{1}{||\mathbf{e}_\Delta||}\mathbf{e}_\Delta^\top D_j \mathbf{w}_1 \\ \phi_2 - \phi_3 + \frac{1}{||\mathbf{e}_\Delta||}\mathbf{e}_\Delta^\top D_j \mathbf{w}_2 \end{pmatrix} = \mathbf{0} \\ \mathbf{0} &= \begin{pmatrix} (\phi_2 - \phi_3)\,||\mathbf{e}_\Delta|| + \frac{\phi_2-\phi_3}{\phi_1-\phi_3}\mathbf{e}_\Delta^\top D_j \mathbf{w}_1 \\ (\phi_2 - \phi_3)\,||\mathbf{e}_{5,4}|| + \mathbf{e}_\Delta^\top D_j \mathbf{w}_2 \end{pmatrix}\end{aligned} \tag{A.4}$$

Subtract 1 from 2:

$$\begin{aligned}0 &= \mathbf{e}_\Delta^\top D_j \left(\mathbf{w}_2 - \mathbf{w}_1 \frac{\phi_2 - \phi_3}{\phi_1 - \phi_3}\right) \\ 0 &= (\lambda_1\mathbf{w}_1 + \lambda_2\mathbf{w}_2 + \mathbf{w}_3)^\top D_j \left(\mathbf{w}_2 - \mathbf{w}_1 \frac{\phi_2 - \phi_3}{\phi_1 - \phi_3}\right)\end{aligned} \tag{A.5}$$

For shorter notation, we write $r_{i,j} = \mathbf{w}_i^\top D \mathbf{w}_j$. Since we require $D \in S_{++}^n$ (see (2.6)), $r_{i,j} = r_{j,i} \wedge r_{i,j} > 0$. Assume we want to find one of the lambda-variables, which we call λ_i, while the other λ will be called λ_k. We then have:

$$0 = \lambda_1 \left(\underbrace{r_{1,2} - \frac{\phi_2 - \phi_3}{\phi_1 - \phi_3} r_{1,1}}_{A_1} \right) + \lambda_2 \left(\underbrace{r_{2,2} - \frac{\phi_2 - \phi_3}{\phi_1 - \phi_3} r_{2,1}}_{A_2} \right) + \underbrace{r_{3,2} - \frac{\phi_2 - \phi_3}{\phi_1 - \phi_3} r_{3,1}}_{B} \tag{A.6}$$

$$\lambda_k = -\frac{B}{A_j} - \lambda_i \frac{A_i}{A_j}$$

If either $A_1 = 0$ or $A_2 = 0$ are zero, the choice of λ_1 or λ_2 (respectively) is arbitrary. The solution for the variable with $A_i \neq 0$ is then $\lambda_i = -\frac{B}{A_i}$. For the general case $A_1 \neq 0$ and $A_2 \neq 0$ we reintroduce the solution of λ_k into the original minimization problem to find the optimal λ_i:

$$\begin{aligned}\lambda_i = & \arg\min_\lambda \lambda (\phi_i - \phi_3) + \lambda_k (\phi_j - \phi_3) + \phi_3 + ||\lambda \mathbf{w}_i + \lambda_k \mathbf{w}_j + \mathbf{w}_3||_{2,D} \\ & \arg\min_\lambda \lambda (\phi_i - \phi_3) + \left(-\frac{B}{A_j} - \lambda \frac{A_i}{A_j} \right) (\phi_j - \phi_3) + \phi_3 + \left|\left| \lambda \mathbf{w}_i + \left(-\frac{B}{A_j} - \lambda \frac{A_i}{A_j} \right) \mathbf{w}_j + \mathbf{w}_3 \right|\right|_{2,D} \\ & \arg\min_\lambda \lambda \left(\phi_i - \phi_3 - \frac{A_i}{A_j} (\phi_j - \phi_3) \right) - \frac{B}{A_j} (\phi_j - \phi_3) + \phi_3 + \left|\left| \lambda \underbrace{\left(\mathbf{w}_i - \frac{A_i}{A_j} \mathbf{w}_j \right)}_{\mathbf{z}_1} + \underbrace{\mathbf{w}_3 - \frac{B}{A_j} \mathbf{w}_j}_{\mathbf{z}_2} \right|\right|_{2,D}\end{aligned} \tag{A.7}$$

and solve the problem using the optimality condition again

$$\begin{aligned} 0 &= \left(\phi_i - \phi_3 - \frac{A_i}{A_j} (\phi_j - \phi_3) \right) + \frac{1}{||\lambda \mathbf{z}_1 + \mathbf{z}_2||_{2,D}} \mathbf{z}_1^\top D (\lambda \mathbf{z}_1 + \mathbf{z}_2) \\ &= \left(\phi_i - \phi_3 - \frac{A_i}{A_j} (\phi_j - \phi_3) \right) ||\lambda \mathbf{z}_1 + \mathbf{z}_2||_{2,D} + \lambda \mathbf{z}_1^\top D \mathbf{z}_1 + \mathbf{z}_1^\top D \mathbf{z}_2 \end{aligned} \tag{A.8}$$

We again write for a shorter notation $p_{i,j} = \mathbf{z}_i^\top D \mathbf{z}_j$

$$\begin{aligned} & -\left(\phi_i - \phi_3 - \frac{A_i}{A_j} (\phi_j - \phi_3) \right) ||\lambda \mathbf{z}_1 + \mathbf{z}_2||_{2,D} = \lambda p_{1,1} + p_{1,2} \\ & \underbrace{\left(\phi_i - \phi_3 - \frac{A_i}{A_j} (\phi_j - \phi_3) \right)^2}_{t} \left(\lambda^2 p_{1,1} + 2\lambda p_{1,2} + p_{2,2} \right) = \lambda^2 p_{1,1}^2 + 2\lambda p_{1,1} p_{1,2} + p_{1,2}^2 \\ & \lambda^2 \left(p_{1,1} t - p_{1,1}^2 \right) + \lambda \left(2 p_{1,2} t - 2 p_{1,1} p_{1,2} \right) + p_{2,2} t - p_{1,2}^2 = 0 \end{aligned} \tag{A.9}$$

The solution to the quadratic problem is

$$\begin{aligned}\lambda_{1,2} &= \frac{-\left(2p_{1,2}t - 2p_{1,1}p_{1,2}\right) \pm \sqrt{4\left(p_{1,2}t - p_{1,1}p_{1,2}\right)^2 - 4\left(p_{1,1}t - p_{1,1}^2\right)\left(p_{2,2}t - p_{1,2}^2\right)}}{2\left(p_{1,1}t - p_{1,1}^2\right)} \\ &= \frac{-\left(t - p_{1,1}\right)p_{1,2} \pm \sqrt{p_{1,2}t^2 - p_{1,1}p_{1,2}^2t - p_{1,1}p_{2,2}t^2 + p_{1,1}^2p_{2,2}t}}{\left(t - p_{1,1}\right)p_{1,1}} \\ &= \frac{-p_{1,2} \pm \sqrt{\frac{1}{\left(t-p_{1,1}\right)^2}t\left(t - p_{1,1}\right)\left(p_{1,2}^2 - p_{1,1}p_{2,2}\right)}}{p_{1,1}} \\ &= \frac{-p_{1,2} \pm \sqrt{t\frac{p_{1,1}p_{2,2}-p_{1,2}^2}{p_{1,1}-t}}}{p_{1,1}} \\ &= \frac{-p_{1,2} \pm k\sqrt{\frac{p_{1,1}p_{2,2}-p_{1,2}^2}{p_{1,1}-t}}}{p_{1,1}}\end{aligned} \tag{A.10}$$

with $k = \sqrt{t}$.

Both solutions need to be checked and the value that minimizes ϕ_4 in Equation (A.1) is taken. λ_k is then easily obtained using Equation (A.6).

Special cases may arise when solving Equation (A.10) unconstrained as Fu et al.already stated:

> *If no root exists, or if λ_1 or λ_2 falls outside the range of $[0, 1]$ (that is, the characteristic direction does not reside within the tetrahedron), we then apply the 2D local solver used in [52] to the faces $\Delta_{1,2,4}$, $\Delta_{1,3,4}$ and $\Delta_{2,3,4}$ and select the minimal solution from among the three.* ([53])

The referenced 2D solver finds the optimal λ for the wave propagation in triangles through solving the following optimization problem:

$$\phi_3 = \min_{\lambda} \lambda\phi_1 + (1-\lambda)\phi_2 + \phi_3 + \sqrt{\mathbf{e}_{4,3}^\top D_j \mathbf{e}_{4,3}}\,, \quad \text{s.t. } \lambda \in [0, 1] \tag{A.11}$$

with

$$\mathbf{e}_{4,3} = \mathbf{v}_3 - (\lambda\mathbf{v}_1 + (1-\lambda)\mathbf{v}_2) = \mathbf{v}_3 - \mathbf{v}_1 + \lambda\left(\mathbf{v}_2 - \mathbf{v}_1\right) \tag{A.12}$$

The derivation is similar to the presented 3D-case.

A.3 Gradient of λ

The optimization is kept as general as possible, by assuming that all variables, except the coordinates of the vertices, are dependant on our exemplary variable ϑ. If we remember the minimizing sign in Equation (A.10), we can state the solution of λ as:

$$\lambda = \frac{-p_{1,2}(\vartheta) + c\, k(\vartheta)\sqrt{\frac{p_{1,1}(\vartheta)p_{2,2}(\vartheta)-p_{1,2}(\vartheta)^2}{p_{1,1}(\vartheta)-t(\vartheta)}}}{p_{1,1}(\vartheta)} \quad \text{s.t. } c \in \{-1,1\} \tag{A.13}$$

In order to derive λ, we recall the definitions of all former temporary shorthand notations and their gradients. Additionally, some of the longer terms also receive their shorthand notation:

$$\begin{aligned}
\mathbf{w}_i &= \mathbf{v}_3 - \mathbf{v}_i & \frac{\partial \mathbf{w}_i}{\partial\vartheta} &= \mathbf{0} \\
r_{i,j}(\vartheta) &= \mathbf{w}_i^\top D(\vartheta)\mathbf{w}_j & \frac{\partial r_{i,j}(\vartheta)}{\partial\vartheta} &= \mathbf{w}_i^\top \frac{\partial D(\vartheta)}{\partial\vartheta}\mathbf{w}_j \\
k_i(\vartheta) &= \phi_i(\vartheta) - \phi_3(\vartheta) & \frac{\partial k_i(\vartheta)}{\partial\vartheta} &= \frac{\partial\phi_i}{\partial\vartheta} - \frac{\partial\phi_3}{\partial\vartheta}
\end{aligned} \tag{A.14}$$

$$\begin{aligned}
A_i(\vartheta) &= k_1(\vartheta)r_{i,2}(\vartheta) - k_2(\vartheta)r_{i,1}(\vartheta) \\
\frac{\partial A_i}{\partial\vartheta} &= \frac{\partial k_1}{\partial\vartheta}r_{i,2}(\vartheta) + k_1(\vartheta)\frac{\partial r_{i,2}}{\partial\vartheta} - \frac{\partial k_2}{\partial\vartheta}r_{i,1}(\vartheta) - k_2(\vartheta)\frac{\partial r_{i,1}}{\partial\vartheta} \\
B &= k_1(\vartheta)r_{3,2}(\vartheta) - k_2(\vartheta)r_{3,1}(\vartheta) \\
\frac{\partial B}{\partial\vartheta} &= \frac{\partial k_1}{\partial\vartheta}r_{3,2}(\vartheta) + k_1(\vartheta)\frac{\partial r_{3,2}(\vartheta)}{\partial\vartheta} - \frac{\partial k_2}{\partial\vartheta}r_{3,1}(\vartheta) - k_2(\vartheta)\frac{\partial r_{3,1}(\vartheta)}{\partial\vartheta} \\
k(\vartheta) &= k_i(\vartheta) - \frac{A_i(\vartheta)}{A_j(\vartheta)}k_j(\vartheta) \\
\frac{\partial k}{\partial\vartheta} &= \frac{\partial k_i}{\partial\vartheta} - \left(\frac{\frac{\partial A_i}{\partial\vartheta}A_j(\vartheta) - A_i(\vartheta)\frac{\partial A_j}{\partial\vartheta}}{A_j(\vartheta)^2}k_j(\vartheta) + \frac{A_i(\vartheta)}{A_j(\vartheta)}\frac{\partial k_j}{\partial\vartheta}\right) \\
t(\vartheta) &= k(\vartheta)^2 \\
\frac{\partial t}{\partial\vartheta} &= 2k\frac{\partial k}{\partial\vartheta}
\end{aligned} \tag{A.15}$$

$$
\begin{aligned}
\mathbf{z}_1(\vartheta) &= \mathbf{w}_i - \frac{A_i(\vartheta)}{A_j(\vartheta)}\mathbf{w}_j \\
\frac{\partial \mathbf{z}_1}{\partial \vartheta} &= \mathbf{w}_i - \left(\frac{\frac{\partial A_i}{\partial \vartheta} A_j(\vartheta) - A_i(\vartheta)\frac{\partial A_j}{\partial \vartheta}}{A_j(\vartheta)^2} \right) \mathbf{w}_j \\
\mathbf{z}_2(\vartheta) &= \mathbf{w}_3 - \frac{B(\vartheta)}{A_j(\vartheta)}\mathbf{w}_j \\
\frac{\partial \mathbf{z}_2}{\partial \vartheta} &= \mathbf{w}_3 - \left(\frac{\frac{\partial B}{\partial \vartheta} A_j(\vartheta) - B(\vartheta)\frac{\partial A_j}{\partial \vartheta}}{A_j(\vartheta)^2} \right) \mathbf{w}_j \\
p_{i,j}(\vartheta) &= \mathbf{z}_i^\top(\vartheta) D(\vartheta) \mathbf{z}_j(\vartheta) \\
\frac{\partial p_{i,j}}{\partial \vartheta} &= \frac{\partial \mathbf{z}_i^\top}{\partial \vartheta} D(\vartheta)\mathbf{z}_j(\vartheta) + \mathbf{z}_i^\top(\vartheta)\frac{\partial D}{\partial \vartheta}\mathbf{z}_j(\vartheta) + \mathbf{z}_i^\top(\vartheta) D(\vartheta)\frac{\partial \mathbf{z}_j}{\partial \vartheta}
\end{aligned}
\tag{A.16}
$$

Additionally we introduce a shorthand notation for the fraction of the square root in (A.13):

$$
\begin{aligned}
u(\vartheta) &= p_{1,1}p_{2,2} - p_{1,2}^2 \\
\frac{\partial u}{\partial \vartheta} &= \frac{\partial p_{1,1}}{\partial \vartheta} p_{2,2}(\vartheta) + p_{1,1}(\vartheta)\frac{\partial p_{2,2}}{\partial \vartheta} - 2p_{1,2}\frac{\partial p_{1,2}}{\partial \vartheta} \\
v(\vartheta) &= p_{1,1} - t \\
\frac{\partial v}{\partial \vartheta} &= \frac{\partial p_{1,1}}{\partial \vartheta} - \frac{\partial t}{\partial \vartheta}
\end{aligned}
\tag{A.17}
$$

Deriving Equation (A.13) using the derivation of all shorthand variables yields the gradient of λ:

$$
\begin{aligned}
\lambda(\vartheta) = \quad & \tfrac{p_{1,2}(\vartheta) + c\, k(\vartheta)\sqrt{\frac{u(\vartheta)}{v(\vartheta)}}}{p_{1,1}(\vartheta)} \\
\tfrac{\partial \lambda}{\partial \vartheta} = \quad & \tfrac{1}{p_{1,1}(\vartheta)^2}\Bigg[\left(\tfrac{\partial p_{1,2}}{\partial \vartheta} + c\, \left(\tfrac{\partial k}{\partial \vartheta}\sqrt{\tfrac{u(\vartheta)}{v(\vartheta)}} + \tfrac{1}{2}\sqrt{\tfrac{v(\vartheta)}{u(\vartheta)}}\, \tfrac{\frac{\partial u}{\partial \vartheta} v(\vartheta) - u(\vartheta)\frac{\partial v}{\partial \vartheta}}{v(\vartheta)^2} \right) \right) p_{1,1}(\vartheta) \\
& - \left(p_{1,2}(\vartheta) + c\, k(\vartheta)\sqrt{\tfrac{u(\vartheta)}{v(\vartheta)}} \right) \tfrac{\partial p_{1,1}}{\partial \vartheta} \Bigg]
\end{aligned}
\tag{A.18}
$$

A.4 Gradient-Approximation of the Diffusion Tensors

We want to find ∇D given D on an unstructured grid. We know that for any closed domain $\omega \subset \Omega$, it holds that

$$
\int_\omega \nabla D(\mathbf{x})\,\mathrm{d}x = \int_{\partial\omega} D(\mathbf{x})\,\mathrm{d}x
$$

Assume our domain of interest ω encompasses exactly one element j. Since we assume constancy of gradient of diffusion tensors $\forall \mathbf{x} \in \omega : \nabla D(\mathbf{x}) = \nabla D_j$ inside one element, we can say that

$$|\omega|\,\nabla D_j = \int_{\partial\omega} D(\mathbf{x})\,\mathrm{d}x$$

Our elements are not arbitrary domains, but are either triangles ($d = 3$) or tetrahedra ($d = 4$), with a finite set of linear lines/faces $\partial\omega_i$:

$$|\omega|\,\nabla D_j = \sum_{i=1}^{d} \int_{\partial\omega_i} D(\mathbf{x})\,\mathrm{d}x$$

In order to compute the integral on the r.h.s., we need to define a proper function that approximates $D(\mathbf{x})$ on the faces. For our experiments, we computed the mean of elements sharing the face.

There are many sophisticated solutions to this problem, but we used the simple assumption that $D(\mathbf{x})$ is the constant mean on each face/line of all elements bordering the line/face. We use a von-Neumann boundary condition with zero flux outside of our domain for all surface faces $\omega_i \in \partial\Omega$.

B List of Publications

Over the whole course of my PhD research from 2017 to 2021, the following publications — all peer-reviewed, or in the process of being reviewed — have been made, including me as an author:

T. Grandits and T. Pock. Optimizing Wavelet Bases for Sparser Representations. In M. Pelillo and E. Hancock, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science, pages 249–262, Cham, 2018. Springer International Publishing

T. Grandits, K. Gillette, A. Neic, J. Bayer, E. Vigmond, T. Pock, and G. Plank. An inverse Eikonal method for identifying ventricular activation sequences from epicardial activation maps. *Journal of Computational Physics*, 419:109700, October 2020, `http://www.sciencedirect.com/science/article/pii/S0021999120304745`

T. Grandits, S. Pezzuto, J. M. Lubrecht, T. Pock, G. Plank, and R. Krause. PIEMAP: Personalized Inverse Eikonal Model from Cardiac Electro-Anatomical Maps. In E. Puyol Anton, M. Pop, M. Sermesant, V. Campello, A. Lalande, K. Lekadir, A. Suinesiaputra, O. Camara, and A. Young, editors, *Statistical Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges*, Lecture Notes in Computer Science, pages 76–86, Cham, 2021. Springer International Publishing

J. M. Lubrecht, T. Grandits, A. Gharaviri, U. Schotten, T. Pock, G. Plank, R. Krause, A. Auricchio, G. Conte, and S. Pezzuto. Automatic reconstruction of the left atrium activation from sparse intracardiac contact recordings by inverse estimate of fibre structure and anisotropic conduction in a patient-specific model. *EP Europace*, 23(Supplement_1):i63–i70, March 2021, `https://doi.org/10.1093/europace/euaa392`

T. Grandits, S. Pezzuto, F. S. Costabal, P. Perdikaris, T. Pock, G. Plank, and R. Krause. Learning Atrial Fiber Orientations and Conductivity Tensors from Intracardiac Maps Using Physics-Informed Neural Networks. In D. B. Ennis, L. E. Perotti, and V. Y. Wang, editors, *Functional Imaging and Modeling of the Heart*, Lecture Notes in Computer Science, pages 650–658, Cham, 2021. Springer International Publishing

T. Grandits, A. Effland, T. Pock, R. Krause, G. Plank, and S. Pezzuto. GEASI: Geodesic-based Earliest Activation Sites Identification in cardiac models. *International Journal for Numerical Methods in Biomedical Engineering (accepted for publication)*, February 2021, `http://arxiv.org/abs/2102.09962`. arXiv: 2102.09962