Reconstructing the electrical dynamics on the heart

Dissertation
For the award of the degree
"Doctor rerum naturalium"
of the University of Göttingen

within the doctoral program
Göttingen Graduate Center for Neurosciences,
Biophysics, and Molecular Biosciences (GGNB),
Physics of Biological and Complex Systems (PBCS)

by
Baltasar Rüchardt
from
Starnberg, Germany

Göttingen, 2022

Thesis Committee

apl. Prof. Dr. Ulrich Parlitz,
Biomedical Physics Group,
Max Planck Institute for Dynamics and Self-Organization

Prof. Dr. Florentin Wörgötter,
Institute for the Dynamics of Complex Systems,
Georg-August-Universität Göttingen

Dr. Andreas Neef,
Center for Biostructural Imaging of Neurodegeneration,
Max-Planck-Institute for Experimental Medicine

Members of the Examination Board

Referee: apl. Prof. Dr. Ulrich Parlitz,
Biomedical Physics Group,
Max Planck Institute for Dynamics and Self-Organization

2nd Referee: Prof. Dr. Florentin Wörgötter,
Institute for the Dynamics of Complex Systems,
Georg-August-Universität Göttingen

Further members of the Examination Board:

Dr. Andreas Neef,
Göttingen Campus Institute for Dynamics of Biological Networks,
University of Göttingen

Dr. Viola Priesemann,
Neural systems theory,
Max Planck Institute for Dynamics and Self-Organization

Dr. David Zwicker,
Theory of Biological Fluids,
Max Planck Institute for Dynamics and Self-Organization

Prof. Dr. Stefan Klumpp,
Theoretical Biophysics Group,
Georg-August-Universität Göttingen

Date of oral examination: 2022-07-21

For Mona, who I wish would continue to live in this shared physical world, and for my families for their support.

Abstract

The heart is an electro-mechanical system. In 2010 about 4.25 million people died from cardiac arrhythmias. Immediate treatment in case of fibrillation is to shock the heart electrically with high energy to reset the electrical system. However high energy shocks cause severe side effects: the heart tissue is damaged, patients develop anxiety and panic disorders. To reduce these side effects low energy defibrillation methods are developed in the research group biomedical physics (RGBMP) at the Max Planck Institute for Dynamics and Self-Organization.

Low energy defibrillation is researched in-vivo and ex-vivo using extracted hearts in Langendorff perfusion. In these experiments the current used for defibrillation is drastically reduced, therefore it is important to understand where the current employed interacts with the heart. For this thesis I rebuilt the experimental setups including shock electrodes and ECG-measurement panels of the RGBMP in-silico using the software Gmsh for computer augmented design (and subsequent computational mesh generation). I extracted the anatomical features of the torso, the heart and the heart muscle's fibre orientation from medical image data and I combined both to create geometrical models of the in-vivo and ex-vivo experiments. I implemented a numerical framework to calculate the current flow in these models and conducted simulation studies together with Master student Simon Wassing.

We found out that the ratio of the total current that interacts with the heart is between 8 % and 15 % of the total current depending on the heart size. Also the positioning of the heart with respect to the electrodes changes the current. We found out that by lowering the heart from a centralised position of the heart between the electrodes in z direction the current flowing through the heart can be increased by up to 20 %.

Patients with a high risk of cardiac fibrillation usually are under pharmaceutical treatment. If this fails ablation therapy is applied. In ablation therapy parts of the heart are burned either by heat or by cold to stabilize the electrical system in the heart to reduce the probability of arrhythmias. A method called inverse ECG or ECG imaging is researched worldwide to provide information about the properties of the heart tissue to identify the regions to burn. I extended the numerical framework with a method for cardiac electric dynamics, potential reconstruction, diffusion of the potential into the vicinity of the bath and ECG signal integration.

The source of the ECG signals is a potential distribution at the outside of the heart. They are referenced against other ECG electrodes to handle that fact that potentials are only defined up to a constant. This can be interpreted as the ECG signal being referenced against a spatial mean. I have found that the spatial mean adds fluctuation to an ECG electrode's signal that is caused by the reference electrodes in a simplified model. To avoid these fluctuations I developed a concept which references against a temporal mean. Comparison of the signals in the simple model shows that the time referenced ECG follows the (in the simulation available) source of the ECG signal better than the spatially-referenced signal.

In the context of inverse ECG the reconstructed source has to be compared to the true source to verify the reconstruction. I showed in a simplified model that the Euclidean distance measure quantifies the distance between states very close in time as large and I have proposed an alternative time-based distance measure. First results show that the time-based distance measure I used based on a model for an excitable cell do not provide a better distance measure but in the corresponding outlook I conclude that future work should try employing oscillator model equations.

Lastly I found from personal experience the lack of a concept how to think of scientific work and classified it as completing scientific activities. I developed a file system structure that reflects this in order to foster reproducibility and traceability of scientific activities.

Table of contents

Chapter 1

Introduction

Research on the heart is done all over the world and organised internationally [1]. The variety of research topics is almost unlimited and reaches from computational research on physics of the whole organ [2] to effects of the COVID-19 virus onto the heart [3] to patching damaged heart tissue with something that resembles plasters for the heart in a research field called cardiac tissue engineering [4]. It will hardly be necessary explaining the necessity of this research in light of cardiac vascular disease being related to about 32 % i.e. to about 17.9 million of global death in 2019 as stated by the WHO [5, 6].

Along these cases one large part is related to ventricular fibrillation, a state in which the heart cells stop beating coordinately with grave consequence. In this fibrillation state the heart loses its ability to pump with the result that the brain is no longer supplied with oxygen as blood stops flowing, leading to death within minutes [7]. Treatment in this case is to try to terminate the state with an electrical shock [8] in order to help the the heart to re-coordinate its cells. The corresponding device is called *defibrillator*. It becomes more and more common to make defibrillators easily accessible. For example, in the German city of Bochum from 2004 to 2009 public accessible defibrillators were installed in the municipal area [9]. Furthermore, events are simulated in which drones deliver defibrillators to patients in northern Germany [10] or web pages and apps are designed to find the next defibrillator [11].

Risk factors to develop fibrillation like sleep apnea, increased age or drinking alcohol are known and health authorities provide means for a self-evaluation of the risk [12–16]. For patients with a high risk of fibrillations defibrillators may be implanted, then called *implantable cardioverter defibrillator* (ICD) [17, 18]. As important as it is, defibrillation comes with side effects including panic attacks due to delivery of shocks while the patient was conscious [19, 20] and an increased risk of mortality after application of shocks [21].

Research on defibrillation strategies is ongoing, both trying to better understand the underlying mechanisms [22–25] and researching new defibrillation strategies [26, 27]. Research includes animal experimentation both in-vivo and with explanted hearts from previously euthanised animals (ex-vivo). One part of the research conducted in the scope of this thesis was to numerically investigate the electrical current flow in these experiments.

Ventricular fibrillation is not only treated immediately. Part of the treatment in reoccurring fibrillation is prevention of the uncoordinated state. In addition to pharmaceutical treatment — in case it is not effective or the medication is not tolerated [28] —, *ablation* is a treatment that has been proven to prevent the onset of a fibrillation [29]. The idea is simple: heart tissue that causes the fibrillation is 'deactivated' by burning it with either heat (radiofrequency ablation) or freezing (cryoablation) [30]. However it is difficult do identify the corresponding heart tissue parts. A complex invasive procedure which physicians call *electrophysiological study* is carried out to find the respective regions [31–34] by triggering fibrillation or coordinated heart beats and based on the observations deducing which regions cause the de-coordination of the cell contraction and the subsequent loss of functionality.

Later in this thesis I will introduce electrical coupling between heart tissue cells as mechanism for coordination and it will identify concomitant changes of electrical potentials (with respect to an arbitrary reference) to precede contraction. These changes in the electrical potentials of the heart cells are the source of the electrocardiogram (ECG), a standard diagnostic tool of cardiology [35, 36]. The signal is used to obtain information about heart tissue regions non-invasively in a research branch called electrocardiographic imaging [37]. It has risen a lot of interest over the last decade, see for example the review by Yoram Rudy [38] or more recent reviews of machine learning techniques applied in this field [39, 40]. This imaging technique already has been used to guide ablation in patients [41]. It reconstructs the electrical potential distribution on the heart surface based on the electrical signal on the body surface. However the heart is also an *excitable medium*, a type of physical system and has a physical state that develops over time. The literature mentioned does not make use of this in the reconstruction process. On the other side to my knowledge little research has been conducted on the data acquisition process used to obtain the ECG data. The second part of my thesis is dedicated to improving the means needed for reconstructing the electrical activity on the heart by (A) revisiting the data acquisition process for the ECG and (B) suggesting a way to incorporate information about the heart being a physical system in a physical state reconstruction process.

The every day scientific work of scientists is full of tasks. So many that scientists have difficulties to keep up with literature [42], scientists tend to work on weekends [43–45]. Concepts how to organise daily scientific work are difficult to find, while an overwhelming number of tools exist. I noticed how difficult it was for me to keep the bigger picture in mind during the every day scientific work. I wondered how can I organise my scientific work in a way that I don't forget it? In the third part of this thesis I will — inspired by the paper by Florian Spreckelsen et al. [46] which I co-authored — suggest a concept how to organise every-day scientific work and data in order to help in contextualisation, improve traceability and ease collaborations and reproducibility.

The scope of this thesis

I conducted research on defibrillation. I built a in-sito equivalent of the experimental setup used in the research group biomedical physics and together with Simon Wassing who was Master student under my supervision I researched how the electrical current behaves in the experimental setup (most of the current avoids the heart in ex-vivo defibrillation experiments). Furthermore, I conducted experiments in a computer model of an in-vivo experiment to understand how different positioning of defibrillators affects the defibrillation current. I extended the numerical setup to also perform cardiac dynamics simulations to help researching the inverse ECG.

In the research on the inverse ECG I have identified two blind spots in current research: How can one *compare* reasonably a reconstructed solution with the signal's true source? Is there a way to make the ECG signal contain *more* information? I will motivate why to research further in these topics and I will suggest two concepts that make use of the underlying system being a time depended system. Two applications will be used to motivate further research and identify problems in the developed concepts.

Lastly I came up in discussions with Alexander Schlemmer, a Postdoc in my group, with a scheme how to organise the daily scientific work in a way it helps remember how the daily task is attached to the scientific research goal — something I tended to forget when focussing on the actual task only.

This makes this thesis and I have ordered it as follows: In the next Chapter, Chapter 2 I will introduce background knowledge needed for the rest of this thesis. I describe the concept how I suggest organising scientific work in the chapter thereafter, Chapter 3. In Chapter 4 I will describe how I obtained the geometrical models needed for the in-sito experiments of the heart, of the heart substructure (fibre orientations) and the position of implantable cardioverter defibrillators and the bone structure from the medical image data. A description of the numerical experiments I have developed will follow thereafter in Chapter 5. Thereafter I will shown in Chapter 6 how defibrillation currents behave in-vivo and ex-vivo and in Chapter 7 I will present the new concept I found as answers when I tried to understand the inverse ECG. Lastly I will conclude in Chapter 8 this thesis and give an outlook on further research.

Chapter 2

Background

This thesis has three main components. First I modelled parts of the ex-vivo experimental setup and an in-vivo defibrillation experiment to understand the current flow in defibrillation experiments and to generate in-silico ECG-Data. In preparation of the numerical experiments I had to process medical data to obtain geometrical models of the tissue, c.f. Section 4, I had to build a geometrical model of the ex-vivo setup and to position the geometrical model of the heart into it, c.f. Section 5.1 and I had to create a finite element mesh used to perform the calculations of the physical equations described in Section 5.1.2.

2.1 The heart coordinates electrically

The heart tissue consists of three layers, the endocardium, the myocardium and the epicardium [47]. The heart muscle encloses four heart chambers. When the heart muscle cells (cardiomyocytes) contract, they reduce the volume of the chambers and blood is pumped through the vessel system of the body. The four chambers are: two large chambers at the lower[1] part of the heart, the ventricles and two smaller chambers at the upper side of the heart, the atria. Figure 2.1a shows these chambers and more anatomical parts of the heart. The ventricular septum separates the left and the right ventricle. The papillary muscle prevents the heart valves (bicuspid valve and tricuspid valve) from opening into the left or right atrium. The inferior vena cava and the superior vena cava are blood vessels bringing blood to the right atria which will then be pumped into the lungs through the right ventricle. Thereafter the blood enters the left part of the heart through the left pulmonary vein where it will be pumped into the body through the left ventricle. The figure also shows the sinus node and the atrioventricular node which play important roles in the conducting system of the heart and will be explained below.

Figure 2.1b provides an overview over the structure of the heart muscle. At the innermost position the endocardium is in contact with the blood inside the heart. Attached to it, there is the myocardium. It consists of the heart muscle cells. It is

[1] Lower in the sense of further away from the blood vessels leaving the heart.

enclosed by the epicardium from the outside. This trinity forming the heart muscle is enclosed by the pericardium, a sack like structure that prevents the heart from contact with surrounding tissue allowing it to beat freely within the restrictions of the pericardial structure [48].

Within the myocardium all cells types are interconnected by *gap junctions* [49]. This means that the cell interior is not self-contained. In particular this means that neighbouring cells are coupled electrically and chemically [50]. Therefore electrical and chemical impulses spread over myocardial tissue without the need of cells transmitting the signal actively. This makes the heart muscle a functional syncytium [51].

The cell interiors are connected and separated from the exterior via the cell membrane. Over this membrane is a potential difference caused by different ion concentration inside and outside the cells [52]. This *transmembrane voltage* of a cardiomyocyte changes rapidly before it contracts [53]. The time development of the transmembrane voltage is called *action potential* of the cell and then one says the cell is *excited*. This excitation spreads due to the syncytial nature of the heart to the surrounding myocardial cells over the tissue. Therefore neighbouring cells contract shortly after each other.

Excitation impulses can be initiated by the heart itself. The sinoatrial node (c.f. Figure 2.1a and Figure 2.1c) describes a region in the heart muscle enclosing the right atrium of the heart holding pacemaker cells that self-depolarise and initiate spreads of excitation. From there it spreads over the right atrium and via specialised cells in the Bachmann's bundle and several other tracks to the left atrium. Finally the excitation wave arrives at the atrioventricular node from where it is routed through the Purkinje fibres to the ventricular muscle. From there it spreads over the ventricles leading to the coordinated contraction from the ventricle tip upwards [52].

So the heart as an organ initiates electrical excitation in the sinoatrial node and transmits it over the heart muscle until finally the contraction follows this excitation waves, i.e. it can be said the heart coordinates electrically.

2.2 Substructure in the heart muscle

The myocardial cells are cylindrical shaped with the short sides touching each other as shown in Figure 2.1d. This organisation of the cells leads to a fibre-like structure. The direction into which the fibres are oriented is called *fibre direction* [55]. At the ends of the cells i.e. into fibre direction there are more gap junctions compared to the other directions [56] which leads to greater conduction velocity of electrical impulses [57]. Furthermore the cardiomyocytes are not only organised in fibre direction but also in sheets [58] with fewer gap junctions in sheet direction compared with the fibre direction but more compared to the sheet-normal direction.

Therefore the heart muscle has an anisotropic microstructure. The local fibre, sheet and sheet-normal direction can be accessed by diffusion tensor magnetic resonance imaging [59]. This will be used in Section 2.5.1 and Section 4.2.3.

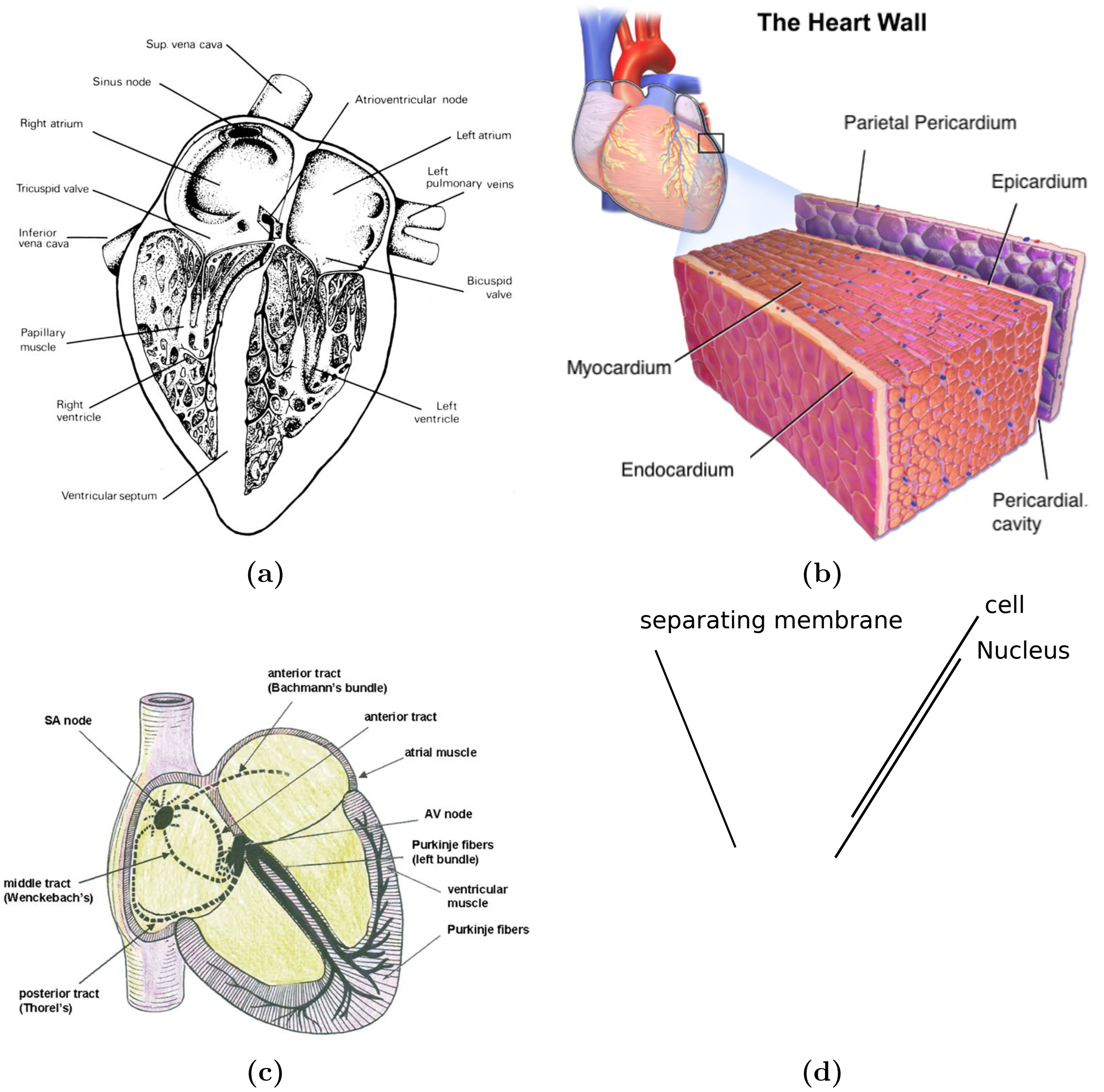


Figure 2.1 – *Figure 2.1a from [47]. Figure 2.1c from [49]. Figure 2.1b from BruceBlaus. When using this image in external sources it can be cited as: Blausen.com staff (2014). "Medical gallery of Blausen Medical 2014". WikiJournal of Medicine 1 (2). DOI:10.15347/wjm/2014.010. ISSN 2002-4436. (https://commons.wikimedia.org/wiki/File:Blausen_0470_HeartWall.png), „Blausen 0470 HeartWall", Labelling adapted by B. Rüchardt, https://creativecommons.org/licenses/by/3.0/legalcode. Figure 2.1d adapted from [54].*

2.3 The loss of coordination: fibrillation

One part of Section 2.1 describes how an electrical impulse triggering contraction comes from the sinoatrial nodes travels through the Purkinje fibres to the heart tip. From there the impulse spreads over the tissue. The cells that were excited contract shortly after — usually originating at the bottom of the heart and moving into the directions of the blood vessels to pump blood into the blood circulatory system.

In normal heart beat the electrical impulse resembles a plane wave travelling from the heart tip upwards. This organised pattern may however break into tachycardia and fibrillation [60]. Tachycardia is associated with the electrical signal travelling in a spiral like pattern [61]. As the mechanical contraction follows the electrical excitation, the pumping is no longer optimised from the heart tip to the outlet into the blood system. In this state the pumping functionality remains, however reduced. This is different for fibrillation: the spiral waves break up forming much more complex spatio-temporal patterns in and on the myocardium [62] and the heart looses its pumping functionality leading to death within minutes.

Immediate clinical treatment comprises a electrical shock trying to reset the heart into a uniform electrical state [63].

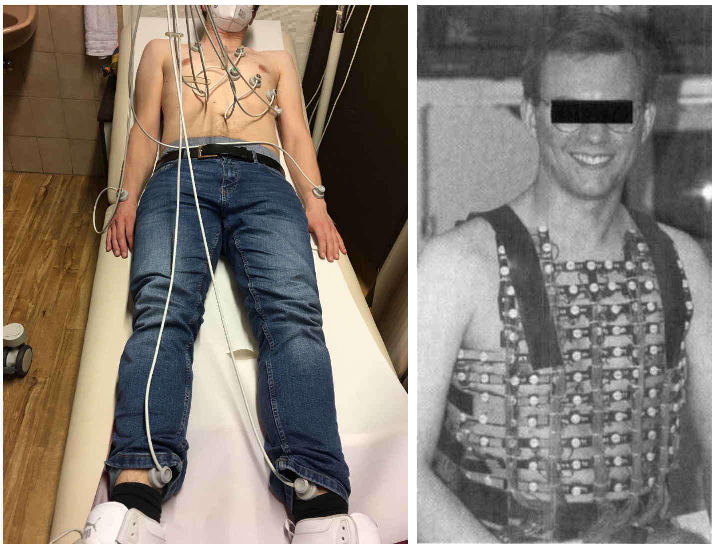
2.4 Accessing the electrical state of the heart

The electrical state of the heart is accessed clinically most commonly with the electrocardiogram (ECG). The ion concentrations outside the heart produce an electrical potential difference. The excitation waves travelling over the heart change this field, an effect that is measurable at the body surface [64]. A standard medical procedure is to measure the *12-lead ECG*. Twelve electrodes are positioned onto a patient's body (Figure 2.2a). The electrode signals are referenced against a combination of the other electrodes leading to electrode signals. Enormous amount of literature exist for medical doctors how to interpret specific ECG signals and how to relate them to diseases, e.g. [35, 65, 66].

In addition to this the branch of *electrocardiographic imaging* (ECGI) is gaining more attention. In ECGI as a first step the body surface potentials are recorded with an electrode vest [67] c.f. Figure 2.2b. Based on the body surface potentials, geometry information about the heart position and the vest electrode positions then the underlying potential distribution on the heart surface is calculated [67].

2.5 Mathematical modelling of the electrical activity of the heart

The principal model to describe heart tissue dynamics is the bidomain model,



(a) (b)

Figure 2.2 – *Figure 2.2a: The author while recording the 12-lead ECG. The electrodes are placed at specific positions on the body. The reference for the electrode measurements are combinations of the other electrodes. Figure 2.2b from [68].*

$$\xi C_m \frac{\partial v_m}{\partial t} + \xi I_{\text{local}} = \nabla \sigma_i \nabla v_m + \nabla \sigma_i \nabla \phi_e, \tag{2.1a}$$

$$\nabla \sigma_i \nabla v_m + \nabla (\sigma_i + \sigma_e) \nabla \phi_e = 0, \tag{2.1b}$$

with v_m being the transmembrane voltage, ϕ_e being the extracellular potential, σ_i being the conductivity tensor giving the conductivity inside the heart cells, σ_e the conductivity in the domain outside the heart cells and I_{local} the current per unit cell membrane through the cell membrane. The value of C_m gives the capacitance of the cell membrane and ξ the ratio membrane to area [69]. There are models to describe the term I_{local}. Overviews are given in reviews, e.g. in [70, 71]. Usually the publications give derivative of the voltage with respect to time,

$$\begin{aligned} \frac{\mathrm{d}v}{\mathrm{d}t} &= f(v, \vec{w}) \\ \frac{\mathrm{d}\vec{w}}{\mathrm{d}t} &= g(v, \vec{w}), \end{aligned} \tag{2.2}$$

with f, g being arbitrary functions and $\vec{w}$ being a vector of helper variables that determine the local current behaviour. These relate to the I_{local} term from the bidomain equations, Equations 2.1 via [69]

$$I_{\text{local}} = -C_m \frac{\mathrm{d}v}{\mathrm{d}t} - I_{\text{app}}, \tag{2.3}$$

where C_m is the membrane capacitance as above and I_{app} is an externally applied current.

There are two cases to consider regarding the boundary conditions of the heart. Either the heart is treated as isolated, i.e. no currents are leaving the heart or coming into the heart over the boundary leading to

$$\vec{n}(\sigma_i \nabla v_m + \nabla \sigma_i \nabla \phi_e) = 0 \tag{2.4}$$

$$\nabla \sigma_e \nabla \phi_e = 0 \tag{2.5}$$

on the boundary of the heart, ∂H, where $\vec{n}$ is the outwards pointing normal vector of the boundary of the heart.

For the case that the heart is coupled to surrounding tissue following [69] the boundary conditions become

$$\begin{aligned} \vec{n}\sigma_e\phi_e &= \vec{n}\sigma_t\phi_t, \\ \phi_e &= \phi_t \end{aligned} \tag{2.6}$$

on the boundary between heart and surrounding volume with σ_t and ϕ_t representing the conductivity and the potential in the surrounding volume, respectively, and $\vec{n}$ defined as before.

If the conductivity tensors σ_i and σ_e depend on each other linearly, the bidomain equations can be simplified to the *monodomain equation.* The monodomain equation only depends on the transmembrane voltage v_m and therefore can be solved numerically much more easily. It reads

$$\frac{\lambda}{1+\lambda} \nabla(\sigma_i \nabla v_m) = \xi C_m \frac{\partial v_m}{\partial t} + \xi I_{\text{local}} \tag{2.7}$$

where λ is a constant scalar factor that gives the linear dependence of the conductivity tensors onto each other,

$$\sigma_e = \lambda \sigma_i.$$

2.5.1 Anisotropic conductivities

The heart muscle conductivity is highly anisotropic. The reason for this is the organisation of the cells in the heart muscle tissue; the cells are organised in fibres with higher conductivity in fibre direction than perpendicular to it [56]. In addition, the fibres are organised in sheets. Conductivity is highest in fibre direction, lower in perpendicular to the fibre but within the sheet and lowest perpendicular to the sheet [69].

The fibre direction is not constant but varies among the heart, while the fibre, sheet and sheet-normal conductivities are considered constant in this thesis. The local fibre, sheet and sheet-normal direction is measured via diffusion tensor MRI [72]. Locally the conductivity tensor can be given in the coordinate system formed by the fibre, sheet and sheet-normal basis vectors, a_l, a_s, a_n multiplied by the corresponding conductivity value $\sigma_{l,i/e}$, $\sigma_{s,i/e}$, $\sigma_{n,i/e}$ of the fibre direction (subscript l), the sheet direction (subscript s) or the sheet-normal direction (subscript n):

$$\sigma_{\text{local, }i/e} = \begin{pmatrix} \sigma_{l,i/e} & 0 & 0 \\ 0 & \sigma_{s,i/e} & 0 \\ 0 & 0 & \sigma_{n,i/e} \end{pmatrix}. \tag{2.8}$$

This conductivity tensor is given in the local coordinate system with the basis a_l, a_s and a_n. The conductivity tensor used in the equation is given in the global coordinate system, i.e. the local conductivity tensor has to be spatially transformed into the global coordinate system. This is done via a mapping A from the global x, y, z coordinate system onto the (local) fibre, sheet and sheet normal coordinate system for each point in space. This mapping is obtained for example via imaging techniques or by heuristics. With the mapping known, the conductivities given in the local fibre directions coordinate system can be transformed into the global coordinate system via

$$\sigma_i = A\sigma_{\text{local, }i}A^T, \tag{2.9}$$

$$\sigma_e = A\sigma_{\text{local, }e}A^T. \tag{2.10}$$

2.6 The Finite Element Method

Throughout this thesis, the finite element method was used to spatially discretise Laplace-like derivatives. The finite element method approximates the real solution by a finite set of basis functions forming the basis of a suitable function space. Solving the mathematical problem reduces then to finding the coefficient for the basis functions of the solution by solving a system of linear equations. In contrast to finite difference methods the finite element method approximates the solution in the *whole* domain and not at the centres of voxel that form the discretisation. Two steps are necessary to employ the finite element method: (A) reformulate the partial differential equation (PDE) problem as variational problem (B) discretise the problem's domain. Figure 2.3 illustrates this concept.

There is a large amount of literature that covers the method from very theoretical to very applied, e.g. [73–75]. In the following sections I will present the process to obtain the system of linear equations.

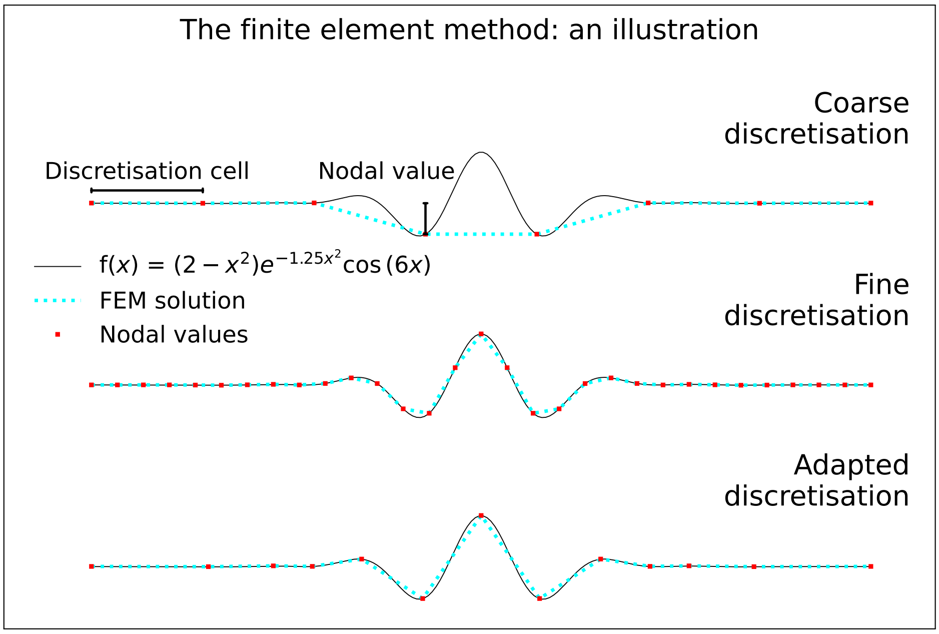


Figure 2.3 – ***Illustration of the idea behind the finite element method and the effect of spatial discretisation on the approximated solution***: *The solution $f(x)$ of a partial differential equation (PDE) is approximated by* ***interpolation functions*** *between the nodal values from a* nodal basis *(x coordinates of the red squares in the figure). The nodal values are obtained by solving a system of linear equations. It is set up by applying variational calculus to the original problem to form the* weak *formulation of the problem followed by employing the discretisation of the domain (see text). The interpolation functions are then defined by the nodal values at the boundary the discretisation cell where the interpolation is performed. Note that in the illustration linear interpolation was used. By adding more degrees of freedom within a cell also higher order interpolation polynomials can be used. The interpolation functions are called finite elements [73, p. 57] and should not be confused with the spatial discretisation cells. The position of the points forming the nodal basis is defined by the domain discretisation.*
The effect of spatial discretisation: *The figure shows three exemplary discretisations. Note that the nodal values always represent the true solution almost exactly. Upper graph: a spatial discretisation that is too coarse will not reproduce features of the true solution. A finer spatial discretisation is able to reproduce these features (centre graph). Also is a problem-adapted spatial discretisation (lower graph): by the right choice of nodal values the number of variables of the linear system describing the approximate solution is small while the accuracy is high. The choice of interpolation functions in the example forces the solution to be continuous on the cell boundaries but it is not differentiable. There are other finite elements that suffice this condition.*

2.6.1 Spatial approximation of a function

In the finite element method the solution function u of a PDE is approximated by a finite set of basis functions from a finite dimensional function space, i.e.

$$u \approx u_h(x) = \sum_i^N c_i \phi_i,$$

where u_h approximates the function u in the N-dimensional function space and with the set of basis functions $\{\phi_i\}$. These basis functions ϕ_i are called finite elements (intuitively 'finite element' is associated with the spatial discretisation, this ambiguity is also discussed in textbooks [73, p. 57]). It is common to discretise the spatial domain into a finite mesh and define the finite elements, i.e. the basis functions of the function space to only be non-zero on the corresponding spatial discretisation cell. Usually the domain is either discretised into simplexes or into hypercubes. Depending on the function space used the continuity of the solution $u_{h(x)}$ over spatial domains varies. Most commonly finite elements are covered in [73, p. 57 ff.].

2.6.2 Variational formulation

The partial differential equation (PDE) has to be converted into the *variational form* to apply the finite element method. This is done by multiplying the PDE with a second function ('test function'). Integrating the equation by parts then gives an equation (the variational form). Variational calculus applied to 'the right' functional shows that the function that solves the variational form independently of the test function also solves the original PDE:

In the following, first the 'right functional' will be defined, minimised and then it will be shown that the parts forming the functional can be identified with terms of the PDE after multiplication with a test function and applying integration by parts. It will be done following [73].

To define the functional, a symmetric positive bilinear form a and a linear form L are defined over a linear space V:

$$\begin{aligned} a: &\quad V \times V \to \mathbb{R}, \\ L: &\quad V \to \mathbb{R}. \end{aligned}$$

One considers the functional $J(v)$ defined as follows:

$$J(v) = \frac{1}{2} a(v, v) + L(v). \tag{2.11}$$

In the next step one considered a function u which minimises the functional. To minimise the functional u has to fulfill the condition $a(u, v) = L(v)$ independently of

v. This can be seen if one consideres a small deviation vt from u and employs the linearity of a and L:

$$\begin{aligned} J(u+tv) &= \frac{1}{2}a(u+tv, u+tv) - L(u+tv), \\ &= \frac{1}{2}\Big(a(u,u) + t\left(a(u,v) + a(v,u)\right) + t^2 a(v,v)\Big) - L(u) - tL(v), \\ &= J(u) + t\left(a(u,v) - L(v)\right) + \frac{1}{2}t^2 a(v,v). \end{aligned} \tag{2.12}$$

It becomes apparent that Equation 2.12 is minimal if the condition $a(u,v) = L(v)$ is true for all v as stated above. To show that every minimum of J fulfils this, one interprets the functional $J(u+tv)$ as function of t. As J is minimal for u the first derivative of J with respect to t has to vanish at u, i.e. at $t = 0$ independently of v. From Equation 2.12 the derivative at u (t=0) is $a(u,v) - L(v) \overset{!}{=} 0$, where the $\overset{!}{=}$ for all v is enforced nu the condition that u mimimises J which is the condition found.

In the next step the bilinear form a and the linear form L are identified with the terms of the variational form. The proof for this can be found e.g. in [73, p. 35]. The result will be shown in the following. As PDE an elliptic boundary value problem of second order (Poisson equation, the standard use case of the finite element method) on a domain Ω is considered:

$$\begin{aligned} \nabla\sigma_2\nabla u + \sigma_1\nabla u = f \quad &\text{in} \quad \Omega, \\ u = 0 \quad &\text{on} \quad \partial\Omega. \end{aligned}$$

This is multiplied with a test function v following the procedure mentioned above — the goal is to obtain a second function representing v in the proof from above to transform the system into a functional — and applying Green's theorem results in

$$\int_\omega \left(\sigma_1\nabla u\nabla v + v\sigma_2\nabla u\right)\mathrm{d}x = \int f v \mathrm{d}x,$$

where already the boundary condition has been applied. The bilinear form a is now defined as the left side of this equation and the linear form L as the right side:

$$a(u,v) := \int_\omega \left(\sigma_1\nabla u\nabla v + v\sigma_2\nabla u\right)\mathrm{d}x, \tag{2.13a}$$

$$L := \int f v \mathrm{d}x. \tag{2.13b}$$

As now the terms from the functional J are defined by Equation 2.13, the minimisation criterium for the variational form can be used to find the solution of the PDE.

This procedure from PDE to variational formulation is sometimes treated as a recipe, e.g. in [75] which focusses on solving PDEs numerically rather than the mathematical aspect.

2.6.3 From variational formulation to a system of linear equations

For numerical simulations, the function space considered in the variational form is spatially discretised[2] with the ansatz from Section 2.6.1. This reduces the continuous infinite function space to a finite space that can be handled numerically. The functional $J(v)$ from Equation 2.11 will be minimised in the spatially discretised function space, i.e. the condition for minimisation is formulated with the discretised solution u_h instead of u:

$$a(u_h, v) = L(v) \qquad \text{for all} \quad v \in V. \tag{2.14}$$

As u_h and v can be represented by their basis function components, $u_h(x) = \sum_i^N c_i\phi_i$ $v(x) = \sum_i^N d_i\phi_i$ (and as this condition has to hold independently of v it is enough to consider the basis functions for v only) the linear and bilinear form can be rewritten as

$$\begin{aligned} a(u_h, \sum_j^N \phi_j) &= \sum_j^N L(\phi_j), \\ \sum_j^N a(u_h, \phi_j) &= \sum_j^N L(\phi_j), \end{aligned}$$

and replacing u_h

$$\sum_j^N a(\sum_i^N c_i\phi_i, \phi_j) = \sum_j^N L(\phi_j) \qquad \Leftrightarrow \tag{2.15}$$

$$\sum_i^N c_i \cdot a(\phi_i, \phi_j) = L(\phi_j) \qquad \text{for} \quad j = 1, 2, \ldots, N\,, \tag{2.16}$$

where in the last step it has been made use of the linearity of a.

This can be written in matrix-vector notation

$$A \cdot \vec{c} = \vec{b}, \tag{2.17}$$

where the entries of the matrix A are given by $A_{i,j} = a(\phi_i, \phi_j)$ and $b_j = L(\phi_j)$ and where c_j is the component of the jth basis function forming the solution u_h. Note

[2] There exist also time-domain finite element approximations e.g. for the solution of wave equations, c.f. [76].

that all entries of the matrix and the vector $\vec{b}$ are known or imposed by a choice of basis functions. This means obtaining the solutions to the partial differential equation reduces to inverting the matrix A in order to obtain the components of the vector $\vec{c}$.

2.6.4 Finite difference approximation

An alternative method to spatially discretise the Laplace operator is the finite difference method. A detailed description of the numerical method can be found in [77, p. 76 ff, p. 104 ff]. In a domain discretised as regular grid in two dimensions, the term $\Delta u(x)$ is approximated by the five-point-stencil by

$$\Delta u(x) \approx \frac{u(x+h,y)+u(x-h,y)+u(x,y+h)+u(x,y-h)-4f(x,y)}{h^2}, \tag{2.18}$$

where h is the step size in the uniformly discretised domain. A partial differential equation becomes a system of ordinary differential equations then which can be used with the known methods.

Chapter 3

Measures for good digital scientific practice

Scientific work has become digital work. However, are scientists prepared for this? My answer will be: No, not in general and in the following I will argue why. In 2006 Neil Beagrie notes not only that "digital resources and data are growing in volume and complexity at a staggering rate" but also concludes that "significant effort needs to be put into developing a persistent information infrastructure for digital materials and into developing the digital curation skills of researchers and information professionals" [78]. Digital curation means "[...] maintaining, preserving and adding value to digital research data throughout its life cycle." as defined by the digital curation centre [79].

On the one hand, this has been done: large research projects like the high-energy physics experiment ATLAS at CERN have developed full data management systems [80]. On the other hand many paper results are not reproducible [81, 82]. Collins and Tabak relate this among other things to poor training in experimental design [83]. As experimental design includes data management therefore also in digital curation.

An interview study conducted at the Purdue University and the University of Illinois at Urbana-Champaign in 2008 and 2009 [84] supports that training of scientists in developing digital curation skills is neglected. It showed that the data curation skills of the students depend on the individual training given by the researcher that supervised them and concludes that there is an increasing need for data information literacy skills. Also today, 10 years later data management or data curation training is not mandatory in general during Bachelor's or Master's degree programmes and students that were supervised by me showed very different data management and curation skills.

This is problematic on the large scale. The United States Congress directed the National Science Foundation ("an independent federal agency to 'promote the progress of science;[...]'"[85]) to *"[...] define what it means to reproduce or replicate a study, explore issues related to reproducibility and replicability across science engineering, and assess any impact of these issues on the public's trust in science."* [86]. It also affects the daily life of scientists: when trying to reuse data, even in the same research group (especially after the person who has conducted the data creation has left the group), when collaborating or when using the script of a colleague.

Based on an initiative of Alexander Schlemmer, we suggest in Florian Spreckelsen et al. [46] an approach to tackle the problem of data being hardly traceable and understandable. We identified the need for a simple, generic and standardized file system layout that is easy to implement. We claim that the 'chain of scientific data management' has three (separate) steps and within these steps we define projects and entries, the latter carrying data and metadata. The implementation proposed to achieve reproducibility and traceability will be described in Section 3.1.

Subsequent to data organisation is data creation process organisation. Data creation processes usually aim at answering a scientific question. Subsequent to the data organisation paper I developed a ready-to-implement concept to contextualise the daily scientific work into the broader picture and to highlight the goal of the current task as the literature covers good scientific practice in general and the problem of reproducibility but does not give concrete implementable concepts. I will present a practical approach that emerged in the aftermath of the work on the paper to keep track of the activities that form the scientific work in Section 3.2.

A condicio sine qua non for reproducibility is to document the parameters of the scientific tasks conducted. Section 3.5 will present the approach used within this thesis based on the CaosDB data model definition [87].

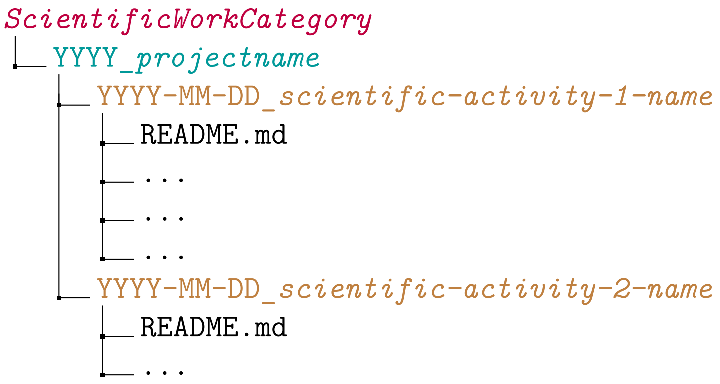
3.1 A file system structure to foster traceability

This section summarises the file system structure published in [46]. To organise scientific data in a standardised fashion we suggest a three-level hierarchy including a standard for naming conventions. The first hierarchy level sorts data in one of four *categories* (see below) which we identified to form the chain of scientific data management. The second level assigns the data to a *project* while the third level is a folder containing the data in an arbitrary substructure and a README.md file containing metadata. The metadata file is expected to hold at least the information *who* is responsible for the data and a description.

The four categories identified are: ExperimentalData, SimulationData, Publications and DataAnalysis. The project folder names are expected to start with the year the project was started followed by the name, i.e. they had the form *YYYY_projectname*. The data level (called *Entry* in the publication) was expected to have the form *YYYY-MM-DD*[*_name*][1] where in case that the exact date is difficult to determine (e.g. if the corresponding data are acquired over multiple days) an approximate date is 'good enough'. This results in the following hierarchy:

1. *Category*: One of ExperimentalData, SimulationData, Publications or DataAnalysis,
2. *Project*: A project name of the form *YYYY_projectname*,

[1] *_name* was not mandatory in the original publication.



```
ScientificWorkCategory
└─ YYYY_projectname
    ├─ YYYY-MM-DD_scientific-activity-1-name
    │   ├─ README.md
    │   ├─ ...
    │   ├─ ...
    │   └─ ...
    └─ YYYY-MM-DD_scientific-activity-2-name
        ├─ README.md
        └─ ...
```

Figure 3.1 – *Generalisation of the file system layout proposed by Spreckelsen et al. in [46] by expanding the first level to an arbitrary category. Each scientific activity is placed in the third level of the three-level hierarchy concept. The first level separates different types of (scientific) work. The second level assigns a scientific activity to a concrete project. The year the project was started is prepended to the project name. The files and folders that contain the work related to a scientific activity are stored in the third level with the date the scientific activity (roughly) was started prepended. It contains a file 'README.md' which provides metadata at will and at least the responsible person and a short description of the scientific activity. Next to the 'README.md' there are other folders ('. . . ' in the graph). They will be described in Section 3.4.*

3. *Entry*: A container with actual results of experiments, simulations or analysis; the name of the corresponding folder is of the form *YYYY-MM-DD_name*,

with the *Entry* level containing a README file, `README.md` that stores a responsible person and description of the data and additionally arbitrary metadata. Note that in the publication was mentioned that more categories or different categories might be used. More details can be found in the paper [46]. A visualisation already including the changes suggested in the next section is provided in Figure 3.1.

3.2 A concept to contextualise daily scientific work

Scientists have to perform many different kind of tasks. They are often from different projects and they are different kind of tasks, often depending on what the next deadline or meeting is. Scientific work would be eased if there was a concept for task management that allows to organise the daily scientific work, that provides an overview over the activities performed and in the best case fosters good scientific practice in order to facilitate traceability and reproducibility of scientific work. I will present a concept that is based on being able to answer the questions "what am I doing" and "what am I doing it for?" whenever one works on a task.

Answering the questions needs to understand the *what* in the question '*what am I doing?*', i.e. the what in the translated question '*what is my current scientific work*'. This 'what' refers to the fundamental unit of the scientific work that is performed.

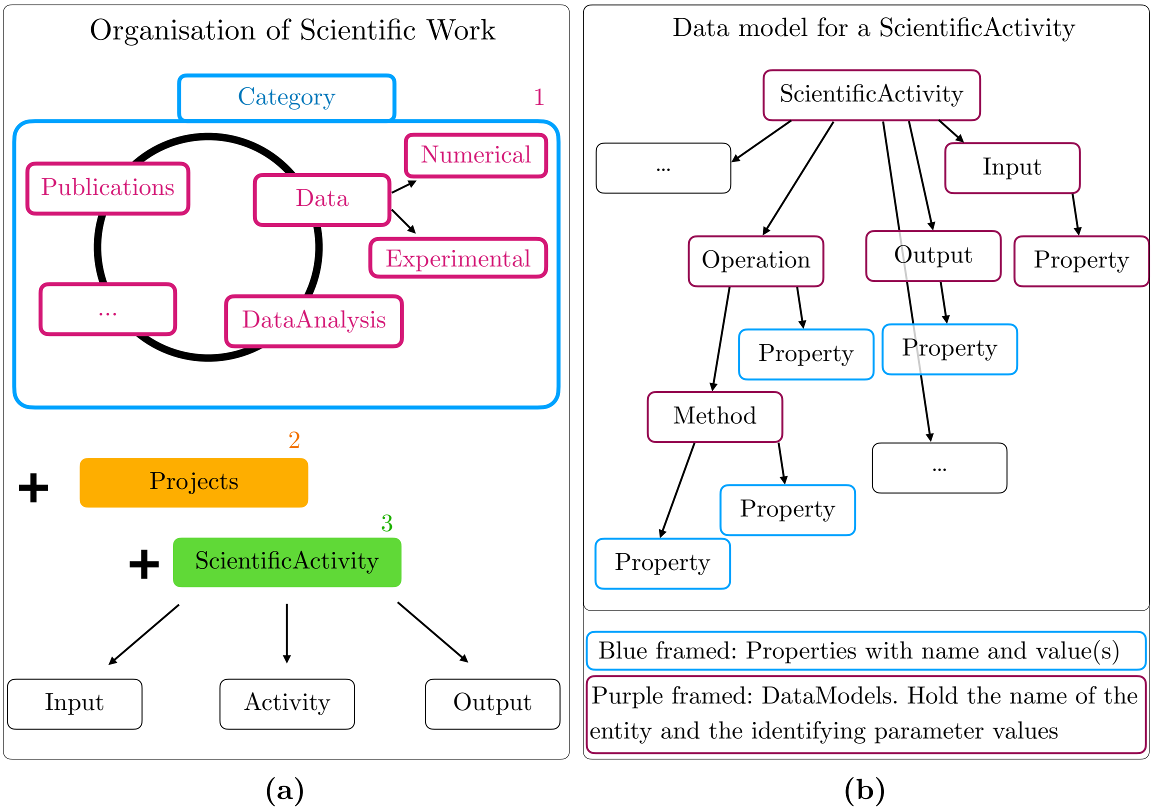


Figure 3.2 – *Figure 3.2a: Visualisation of the three-level hierarchy model proposed as a model to think of scientific work. The first level* category *(1) categorises which type of work the researcher is doing. This differs very much between different research groups. The idea is to answer the question* "what am I doing?" *by the name of the category. The second level answers the question* "what do I do it for?". *Based on the assumption that scientific work is organised in projects. This level assigns the work to a* project *(2). The third level* ScientificActivity *(3) is the activity itself. Further structure for the scientific activity level will be provided in 3.4. The file system layout followed this model. Figure 3.2b shows the approach to represent an abstract scientific activity as a data model. A data model in this context is meant to store information. In this context, the individual tasks and their parameter are the information to store: A scientific activity is composed of entities. These entities are themselves data models and carry properties. The properties are name – value pairs which define the concrete representation of the (abstract) data model. Initiated by Alexander Schlemmer and thereafter together with him we worked on documenting scientific work with this approach.*

I will call it a *scientific activity.* Scientific activities are manifestations of one sort of scientific work like for example data analysis, realisation of a real or numerical experiment, visualisation of data, or research field specific sorts of work (see below for an example used in the course of this thesis). Identifying the sorts of scientific work that one performs will provide a pool of answers to specify an activity. To answer the question '*what am I doing*' then reduces to picking the scientific work category the scientific activity belongs to.

Answering the question "what am I doing it for?" will be done by contextualising the current task to the bigger picture, i.e. to the current goals. The goals are usually organised in projects, the question 'what am I doing this for?' then can be answered by: 'Ah yeah, I wanted to check [. . .] for the project [. . .]'. The current projects create a pool of answers which help to contextualise ones work.

To summarise, this concepts has three components that help a researcher to understand *what* they is doing and what they is doing it *for*:

1. ScientificWorkCategory: Describe the sort of scientific work performed.
2. Project: Scientific work tries to achieve a goal. These goals are organised in projects. Saving the work alongside the goal focusses on why the activity is done.
3. ScientificActivity: a actual realisation of a scientific activity. Each step towards the goal is separated. This forces to think of what he / she is actually doing.

This concept is visualised in Figure 3.2a.

I used the scientific work categories *NumericalExperiments*, *DataAnalysis*, *VisualisationScripts*, *SoftwareDevelopment*, *FemMeshCreation* and *TryingOut* to categorise my scientific work activities in the course of this thesis.

3.3 Complementarity between the previous sections

The previous section and the section before both come up with a three level hierarchy. Both categorise their entity – data or a scientific activity – on the first level, assign their entity to a project and in the last hierarchy level place the entity itself. This similarity invites to combine the technical implementation of a standardised file system structure and the conceptual approach to think about scientific work to

A) Organise scientific activities on the file system similar to how Spreckelsen et al. [46] organise scientific data.

B) Open the *Category* level of the file system layout structure to contain more categories, determined by the need of the research environment.

C) Apply the concept of metadata storage to scientific activities.

It is not an easy task to extend the categories on the file system. The set of categories chosen by a researcher is not predetermined by a physical law. Therefore different researchers may come up with different categorisations for similar data and scientific work categories. But it is important that within one file system structure categories are chosen meaningful and unambiguous.

To the predefined data categories *SimulationData*, *DataAnalysisOutput* (renaming the category *DataAnalysis* from [46]) I added the data categories *FemMeshes*, *Segmentations* and *Visualisations* to the file system. The aforementioned research categories were used to structure scientific activities on the file system. In the scope of this work I used *NumericalExperiments*, *DataAnalysis*, *VisualisationScripts*, *SoftwareDevelopment*, *FemMeshCreation* and *TryingOut* to classify scientific activities.

3.4 Structuring scientific activities

Up to this point I have suggested a concept of scientific work that can be emulated on the file system using the structure proposed by Spreckelsen et al. [46]. I introduced *scientific activities* as smallest entity of scientific work. In this section I will specify scientific activities further.

I define a scientific activity as smallest unit of scientific work. Sometimes it might be that several tasks have to be done to finish one scientific activity. However as long as they cannot stand on their own I don't consider them a scientific activity. As a result one scientific activity is associated with only one scientific task. The activity is defined by the sequence of operations, i.e. sub tasks performed within the activity. E.g. (i) loading input, (ii) applying a filter, (iii) use a model, (iv) transforming data, (v) calculating a quantity, (vi) saving output – usually defined by a computer code script – and possibly by parameters used to adjust the operations. Parameters could be the bandwidth of the filter in the example or the parameters of a model that was utilised. Figure 3.2b shows an overview over the different parts of a scientific activity including a concept that will be presented in the next section.

The most general concept of an activity is: 1. input 2. activity 3. output. For example a data analysis script gets raw data to analyse and operates on them to produce output, the result of the analysis. Note that the input usually is the output of another scientific activity.

This concept of (input), activity and output can be reproduced on the file system below the scientific activity level: an input folder holds links to activities the activity is based on. The output folder where this activity output data is stored in the data structure is linked in the activity folder. For a folder representing a scientific activity this leads to the following file system substructure:

- *Input* [optional]: Links to the input data were stored here
- *Activity*: Realisation of the scientific activity
- *Output*: Link to the activity's output folder

This is visualised in Figure 3.3. The linking ensures that data and scripts are kept separately.

```
YYYY-MM-DD_scientific-activity-1-name/
├── Input/
├── Activity/
├── Output/
└── README.md
```

Figure 3.3 – *Scientific activity folder structure: A* `README.md` *file provides meta information about the activity. The other files in the folder are structured as follows:* `Input`*: Input parameters are stored or created here.* `Preprocessing`*: Scripts doing the preprocessing are stored in this folder.* `Activity`*: Scripts performing the actual activity are stored in this folder, the folder is named to the first level in the hierarchy.* `Output`*: All kind of output files are stored here. Other subprocesses might be added depending on the specific activity.*

In addition I suggest to subclassify an activity itself in *pre_activity*, *activity* and *post_activity*. With this scheme scientific activities can be standardised. A standardised activity script in python is presented in listing 3.1.

3.5 Scientific activity parameter documentation

The last section identified scientific activities as a number of operations and corresponding parameters. Documentation of parameters of the operations forming a scientific activity is crucial for reproducibility and activity documentation. Therefore Alexander Schlemmer and I developed a concept on how to store scientific activity data based on the CaosDB data model framework published by Fitschen et al. [87]. The scientific activity is split into single operations. Each operation may have parameters (see above) that are associated with it. Traceable documentation needs not only to store all parameters but should also include the activity they belong to. This results in a scientific activity model (in analogy to a data model presented by Fitschen et al.) as presented in Figure 3.2b.

3.6 Summary and Discussion

I started with the claim that (in general) scientists are not prepared for their work to have become digital. No wonder in a world where scientific research has become increasingly complex and accessibility to information has reached unprecedented levels. This brought me to a concept where *scientific activities* are the base unit of scientific work and which allowed to structure the daily digital scientific work accordingly.

One step further I reported on how to continue this process to structure the every day scientific work on the scientific activity level. I identified two complementary concepts: the first is to structure the scientific activity in *Input*, *Output* and *Activity*, a concept that I have translated to the organisation of the scientific work on the computer.

```python
import argparse
import GoodScientificPractice as gsp
import utilities as uts

def _activity_arguments():
    parser = argparse.ArgumentParser()
    """ ... add arguments ... """
    return vars(parser.parse_args())

def pre_activity():
    args = _activity_arguments()
    inputs = gsp.InputLogger()
    outputs = gsp.Outputs(args["output_folder"])
    # Mark InputDocumentation for saving
    outputs.add_output("InputDocumentation.txt", gsp.writers.
        obj_as_stringrep(inputs))

    file_path = args["input_files"]
    # log input files
    for filename in file_path:
        inputs.log(filename, "SimulationOutputFile")
    files = [uts.load(fp) for fp in file_path]

    return {"outputs": outputs, "inputs": inputs, "file_content":
         file_content}

def activity(file_content, outputs, **kwargs):
    """... Perform activity ..."""
    return {"outputs": outputs}

def post_activity(outputs, **kwargs):
    """... Save marked outputs ..."""
    outputs.save_current()

def main():
    pre_activity_output = pre_activity()
    activity_output = activity(**pre_activity_output)
    post_activity(**activity_output)

if __name__ == "__main__":
    main()
```

Listing 3.1 – *Minimal Scientific activity code. The activity is split into three parts:* pre_activity, activity *and* post_activity. *Inputs, parameters and outputs are logged with a python package GoodScientificPractice [88] developed as part of this thesis. Parameter documentation strongly builds on work done by Alexander Schlemmer and the CaosDB data model [87].*

The second is to structure a scientific activity *within* itself into *pre_activity*, *activity* and *post_activity*.

Application of these concepts have the potential to increase productivity, traceability of scientific work performed and it did so for me within this thesis. It helped to find and to reuse data and scripts especially between me and the students that I supervised.

This aspects are relevant on the personal level. However it has also the potential to increase the quality of scientific work: I believe that reproducibility and traceability will be eased if a underlying concept is applied compared to an unguided approach in which scientists create their data management concepts on the fly.

The work with the concept of scientific activities and the substructure of them emerged casually the idea of applying the CaosDB data model approach initially intended to document scientific data also to scientific activities.

There is something else that is important to mention: this concept can be used as a guide on how to think of ones everyday scientific work. However I do not want to impress this concept on anyone. As long as a scientist has found their own way through the demanding scientific daily routine and as they is happy with their own way I don't want them to change anything. I put this chapter in this thesis for people that have not yet found their way and are (like I am) persons that prefer to have a framework to conduct their work in.

I believe that consciously separating scientific activities and scientific data will help scientist to order their scientific work.

Chapter 4

Medical Image Processing

In this work I performed simulations based on medical data. In order to obtain simulation meshes for the finite element method, those data had to be preprocessed. This includes:

- Segmentation of the tissue
- Extract surface data for the volumes
- Post-processing the surfaces with meshlab

Medical data included C-arm x-ray machine data, MRI and diffusion MRI data. All data were provided by Dr. Claudia Richter. All data were measured with Göttingen minipigs.

4.1 Tissue segmentation

The process of identifying different regions in medical image data is called segmentation. Segmentation of all medical data was done using 3D slicer [89–92]. The corresponding user guide [93] provides a section about image segmentation and further links to tutorials, which I have been following in this thesis. The program's segmentation module is described in [94]. Medical image data were processed by me, by me with help of Dr. Claudia Richter, and by master student Simon Wassing and bachelor student Benjamin Weiß. All segmentations are based on an intensity / signal value which changes depending on the kind of tissue. Tools in the segmentation module help marking the 3D volume of tissue based on marked tissue cross sections in 2D slices. Many tools are built-in, in addition I used the *Fill between slice* algorithm [95] and the *Grow from seeds* algorithm [96].

4.1.1 C-arm in-vivo data

C-arm data from a Göttingen minipig were provided by Dr. Claudia Richter. A C-arm can produce 3D images using x-rays. The resulting data contain for each voxel the intensity. Variations in intensities between the voxels are caused by different absorption coefficients of the corresponding tissue. Denser tissue absorbs more x-rays. C-arm machines can be used with living animals. The Göttingen minipig of which the scan was taken had a catheter electrode and an implantable cardioverter-defibrillator (ICD) implanted. Segmentation was done together with Dr. Claudia Richter. To handle the large grey-scale space, only voxel in a certain range are displayed with grey scale values, while the rest is black / white. This range of intensities displayed is called *window*, the position on the greyscale scale is called *level*. Figure 4.1 shows how changing these values affects the displayed window (i.e. the image that underlays the segmentation). From Figure 4.2 it can be seen that separating the heart from the surrounding tissue is difficult because the fat tissue surrounding the heart and the heart muscle show the same absorption. It was not possible to properly separate the heart muscle from the blood inside the heart based on the data obtained.

The boundaries of the segmented tissue were exported as .stl surface data for further processing.

4.1.2 MRI Data

The segmentation process of the MRI data is described in the theses of Benjamin Weiß [97] and Simon Wassing [98]. These data were provided by Dr. Claudia Richter, too. The imaging process was conducted on a Göttingen minipig in-vivo. To increase the contrast between blood (in the blood vessels and in the heart chambers) and the heart muscle tissue, a contrast agent was used. The high contrast between blood and surrounding tissue allowed to segment the blood vessels and heart chambers very accurately as can be seen in Figure 4.4a. Benjamin Weiß used this data in combination with the C-arm in-vivo data to overlay the medical images in a process called registration and displayed the resulting data in virtual reality. He successfully combined data from a C-arm scan and an MRI scan to align the heart in an combined image. However there were still errors for other, e.g. the vena cava which did not align after the registration had completed. Registration was done using 3D Slicer. The virtual reality rendering was done in Paraview [99, 100]. The data used were defibrillation currents in the ex-vivo setup (see Section 5.1.3.4). This work was supervised by me, the results in detail can be found in [97].

4.1.3 Diffusion MRI data

The heart muscle is structured in sheets and the sheets are structured in fibres. The fibre and sheet orientation impacts the conductivity in these directions as described at the end of Section 2.5. The diffusion tensor MRI measures the diffusion of water molecules in the volume. The diffusion is highest in direction of the fibres as there are more gap junctions. There are two directions perpendicular to the fibre direction. The direction with the second highest diffusion is within a muscle sheet while the third direction is perpendicular to both, fibre and sheets.

The diffusion rate does not give the conductivity in that direction. Conductivity in the aforementioned directions has to be measured separately and will be employed as describe in Section 2.5.1. The diffusion MRI data were provided by Dr. Claudia Richter. The measurement was conducted ex-vivo. Segmentation was mainly done by Simon Wassing and is described in his master's thesis [98]. He built on work done by Benjamin Weiß. The explanted heart has a very high contrast to the surrounding fluid and therefore segmentation can be done with great detail as can be seen in Figure 4.4b.

Further processing of the segmented tissue needed only the outer most surface of each volume. The boundaries of the segmented tissue were exported as .stl surface data for further processing, the diffusion tensor data were exported as .NRRD files.

4.2 Segmentation data post-processing

The surfaces of the segmented volumes in slicer were exported as .stl surface data. For the case of the diffusion tensor data, it had to be taken care of exporting the surface data and the diffusion tensor data in the same spatial coordinate system. 3D Slicer uses the right-anterior-superior coordinate system [102], i.e. the diffusion tensor data were saved in this coordinate system. This option has to be chosen explicitly on export of the surfaces. The surface data obtained from the segmentation had a comparably high number of faces and fine structures which were not considered necessary for the simulations. To reduce the number of faces and to smooth the surfaces I used meshlab [103] and later gmsh [104]. The tools and procedures used will be covered in Section 4.2.1.

A second post-processing operation necessary was to move and rotate the meshes such that they fit in geometries designed for the numerical experiment, c.f. Section 5.1.1. In the case of the diffusion MRI data, this also included transforming the associated diffusion-MRI tensors indicating the fibre orientation. This process will be covered in Section 4.2.2.

4.2.1 Surface mesh post-processing

Post-processing of the surfaces was done in meshlab [103] and in the course of the thesis in gmsh. Details were reduced as the segmentations weren't done in very high detail and details require a high mesh resolution resulting in long computation duration. This was done manually for the C-arm data using various meshlab filter scripts. For the MRI and diffusion MRI data the meshlab filter *Remeshing: Isotropic Explicit Remeshing* and the smoothing filter *Laplacian Smooth* [105] were used. The first performs operations to improve and remesh a mesh based on the existing mesh reducing the total number of surfaces representing the mesh. The latter smooths edges in the geometry. Remeshing and smoothing was necessary because the finite element solver doesn't converge for volumetric meshes with very low mesh quality. More about mesh quality can be read e.g. in [106] and the references therein. As the segmentation of the in-vivo c-arm data is very rough, the mesh didn't have to be very detailed and a comparably large mesh surface discretisation could be used, as can be seen by comparing Figure 4.3a and Figure 4.3b.

4.2.2 Coordinate transformations

The segmented surfaces were placed into a geometrical model, either a model for the pig body or — in most simulations — in a geometrical model of the experimental setup, c.f. Section 5.1.1. Therefore the coordinates had to be transformed. This was done manually using meshlab. First the necessary angles for rotation and translation were determined by rotating the heart in a way the vector of the aorta leaving the heart muscle points in z-direction, mimicking the position in the experiments. Thereafter the heart was translated to fit the position in the geometrical model, i.e. the heart centre was at $\vec{x}_{\text{centre heart}} = (0, 0, 40)$. The necessary rotations and translations were tested with the outer boundary of the heart muscle and then applied to the hollows in the inner using a filter script and meshlab server to automatically transform all related meshes at once.

Transformation of the diffusion tensor data was implemented by me based on the meshlab filter script. Rotation of the coordinate system also affects the tensor data. A rotation of the image coordinate system represented by the rotation matrix R implies a transformation of the diffusion tensor [107]. This means for the diffusion tensor $D_{i,j,k}$ at voxel with index i, j, k

$$D' = RDR^T,$$

where D' the diffusion tensor in the new coordinate system and R^T represents the transposed matrix R. The diffusion tensors are invariant under translation, i.e. the spatial coordinates could be translated without the need to adapt the diffusion tensors.

4.2.3 From diffusion tensor data to conductivities

The cardiac muscle cells are organised fibres and sheets. The conductivity in fibre direction is larger compared to the conductivity in sheet direction which again is larger to the conductivity perpendicular to the sheet. This information is expected to influence defibrillation currents and will therefore be measured and used in the numerical framework. The diffusion tensor data represent the diffusivity in each direction in the lab coordinate system. In order to obtain the fibre, sheet and sheet-normal directions, the eigenvectors of the diffusion tensor at each pixel have to be calculated for the diffusion tensor of each voxel. Thereafter to obtain the global conductivity tensor following the theory described in Section 2.5, the fibre, sheet and sheet-normal conductivities were taken from literature and are indicated in Table 4.1.

The diffusion tensors of the segmented heart volume showed in about 3 % negative eigenvalues. Diffusion tensors are assumed to be positive definite and negative eigenvalues are explained by noise [109], therefore I considered these values measurement errors and I interpolated the fibre, sheet and sheet-normal vectors of those voxel based

Table 4.1 – *Conductivities. σ_l, σ_t and σ_n are taken from [69], σ_{fat} is taken from [108].*

	σ_l	σ_t	σ_n	σ_{fat}
conductivity [µS/cm]	3000	1000	315.25	333

on the surrounding voxel using the LinearNDInterpolator of scipy [110] if possible. If the voxel's orientation data could not be interpolated because they laid outside the convex hull of the segmented heart tissue, I replaced them by isotropic tensors with the conductivity of fatty tissue. The reasoning behind this is: (A) the diffusion tensor MRI data could be very easily distinguished from the solution outside of the heart as they were measured ex-vivo (B) if the neighbouring voxel does not provide a fibre direction the voxel is not part of the heart muscle tissue (C) as it is not part of the heart muscle tissue but not part of the enclosing solution either it is fat. In addition I implemented gaussian smoothing to reduce the effect of noise.

This gave me direction which I transformed as described in Section 4.2.2 based on the transformation script saved with meshlab in order to fit the final numerical experiment setup, c.f. Section 4.2.2. I then interpolated the fibre orientation data (that are stored on a 3D regular grid) to the centres of the tetrahedral cells forming the volumetric finite element mesh. In the case that the simulation domain contains mesh cells outside the heart boundary, the .stl data from the segmentation were used to set cells outside the heart tissue to be of isotropic conductivity with a conductivity value chosen by the user, i.e. to the conductivity of the surrounding medium.

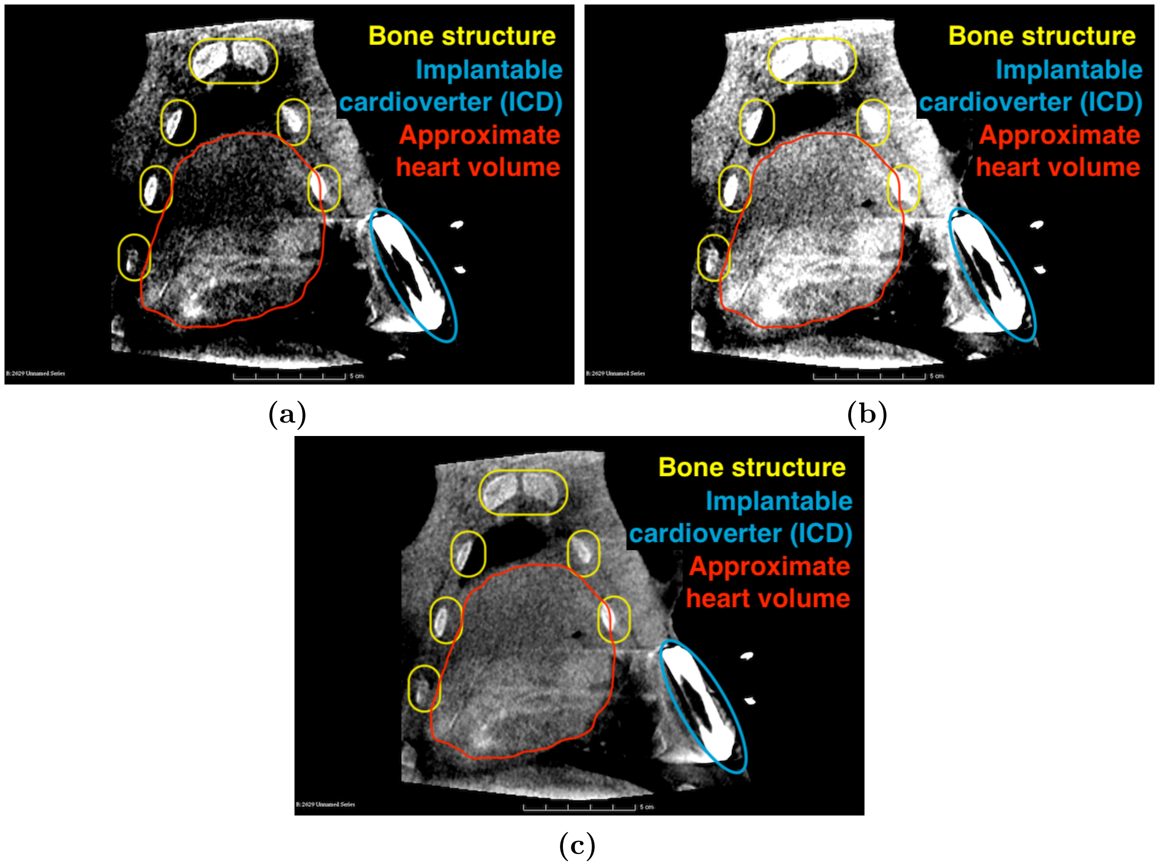
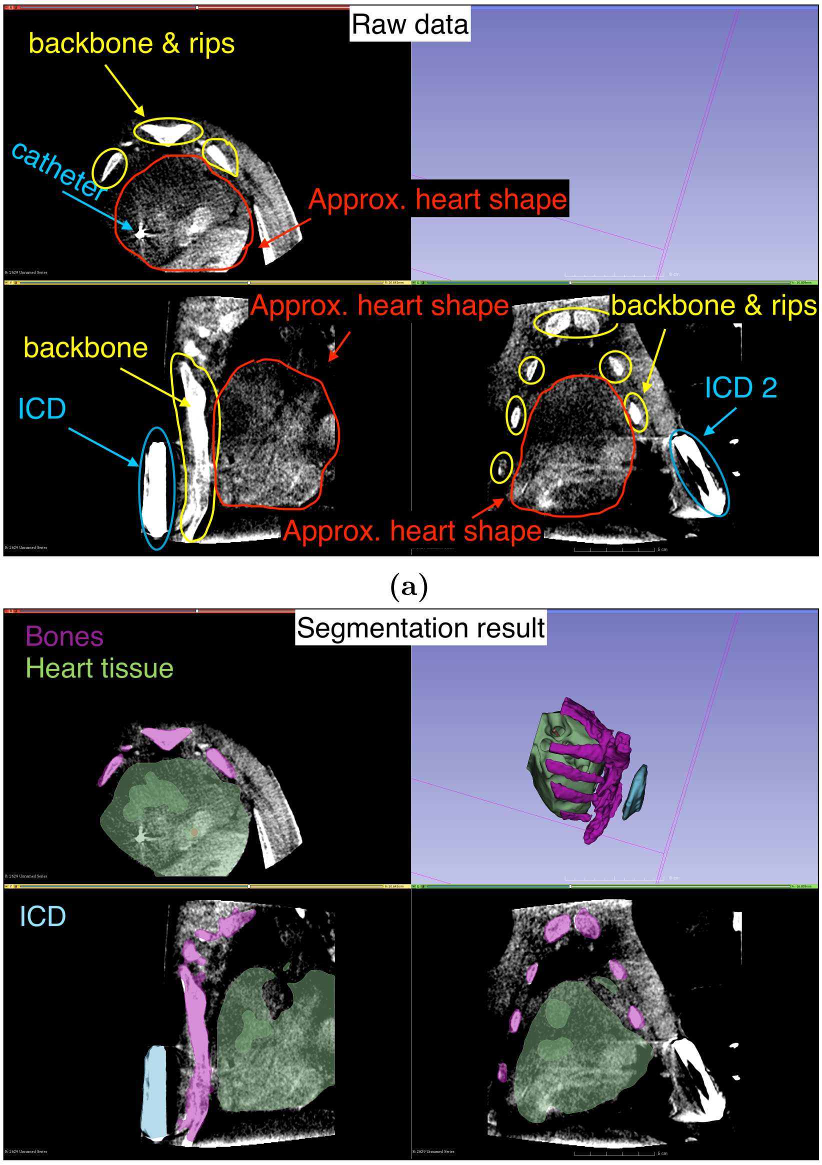


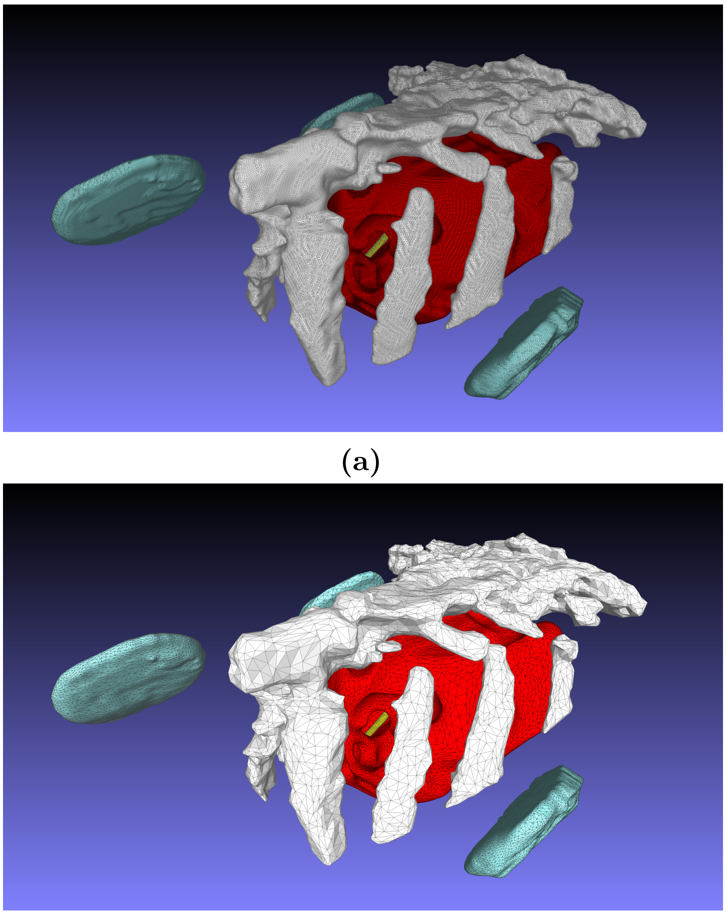
Figure 4.1 – *Effect of changing the window size and the level of the intensities displayed in 3D slicer. Figure 4.1a: Reference image. The heart (centre, in red), cross sections of multiple rips (surrounding the heart left and right, in yellow) and an implantable cardioverter-defibrillator (right in blue) can be seen. Figure 4.1b: The window in which the grey scales are resolved is shifted to lower values changing the level displayed. Intensity values that appeared dark before are now displayed bright. Figure 4.1c: The value level now is kept constant (i.e. the central position of the window in intensity space) but the window size (i.e. the width of the interval displayed) is increased leading to more intensity values being considered displaying the image. Small variations in intensity then are less fine resolved leading to a smaller contrast. The density of the part of the body that is being segmented, the level and the window size are changed in order to make the difference to surrounding tissue as visible as possible. Algorithms help in the segmentation by detecting boundaries or by connecting 2D segmented tissue slices with each other.*



(a)

(b)

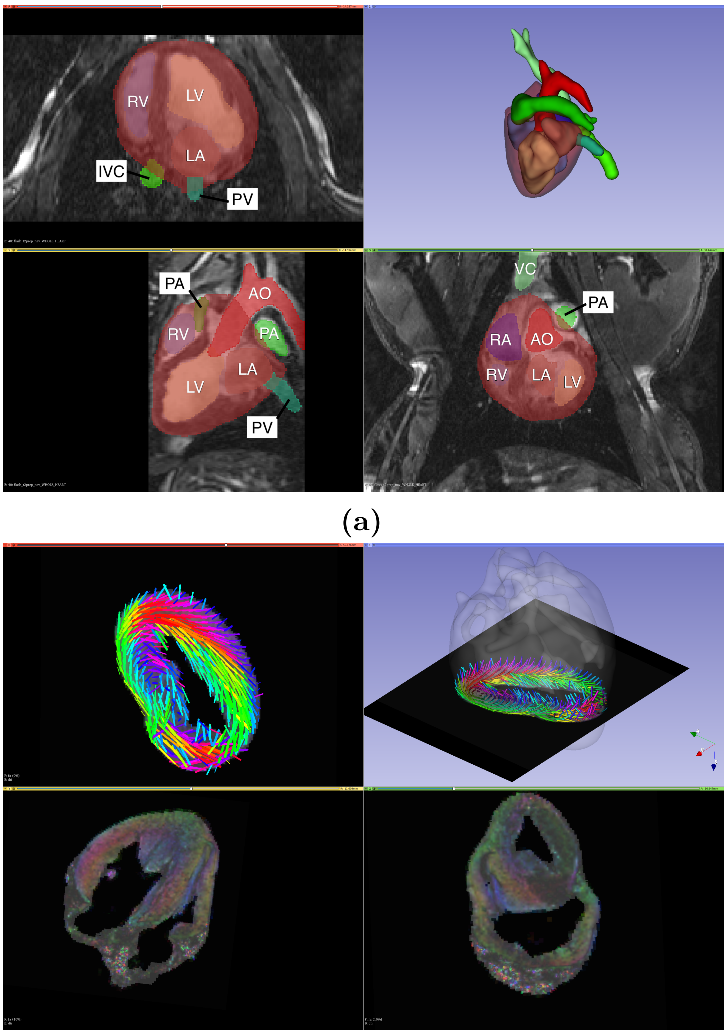
Figure 4.2 – *Medical C-arm in-vivo data from a Göttingen minipig segmentation: Figure 4.2a: Before segmentation. The window and level values are adapted to make the bones and metal visible. The upper left image shows the bone structure and the catheter in the heart. The upper right picture is empty. It will depict the segments in 3D. The lower left shows the spine, an ICD and adumbrates the heart muscle. The lower right picture suggests also a shape of the heart muscle, and contains cross sections of the rips. In addition, a second ICD can be seen on the lower right in that picture. Figure 4.2b: The same picture with segmented materials highlighted. Bone structure is shown in purple, heart tissue in green, the ICD in use in light blue and the catheter in brown (only 3D view). Upper right: 3D view of the segments.*



(a)

(b)

Figure 4.3 – *Medical C-arm in-vivo data from a Göttingen minipig surface meshes: three ICD positions are segmented in blue, the bone structure is coloured white, the heart is coloured in red and inside the heart the catheter can be seen in yellow. Figure 4.3a: Exported surface data from 3D slicer shown in meshlab before surface mesh simplification. Figure 4.3b: The surface meshes after simplification and smoothing in meshlab. Note that the bone structure changes slightly with the remeshing process and that the number of cells decreases.*



(a)

(b)

Figure 4.4 – *Figure 4.4a: Segmentation of the in-vivo MRI data conducted by Simon Wassing and Benjamin Weiß. In data acquisition a contrast agent was added to increase the contrast between blood in the blood vessels and heart chambers and the heart muscle tissue. As a result, the blood is easy to see in the segmentation process. In the image the volume that defines the outer boundary of the volume considered being the heart muscle is overlaid in transparent dark-red. Abbreviations used are LV: left ventricle, LA: left atrium, RA: right atrium, AO: aorta, VC: vena cava, PA: pulmonary artery, IVC: inferior vena cava, PV: pulmonary veins. Simplification of the surface mesh was done during volumetric mesh creation using gmsh, see Section 5.1.2. Figure 4.4b: Fibre orientation data displayed in 3D slicer. The colours indicate the fibre direction based on the colouring explained in [101]. The lower panels show the direction of the fibre orientation in exemplary slices. The upper left panel shows the fibre direction in the slice that is also displayed in the 3D view (upper right) as tubes. Upper right panel: The segmentation of the heart is shown in transparent white. The cross section from the upper left panel is shown including the tubes indicating the fibre orientation at the corresponding voxel.*

Chapter 5

Numerical Experiments

A large part of this thesis was to design numerical experiment setups based on the real experimental setup. Section 5.1.3 describes the geometrical models designed to model real world experiments. The second section of this chapter, Section 5.2 describes the numerical experiments conducted using the numerical setups.

5.1 Numerical experiments design

I have developed several geometrical setups used to conduct numerical experiments. The setup design is done in two steps:

- Creation of a geometrical model
- Spatial discretisation of the model

This process is independent of the actual numerical experiment and will be described in the first two Sections, Section 5.1.1 and 5.1.2. Thereafter Section 5.1.3 introduces the actual setups used. An overview of the base for the modelling process if applicable, the geometrical models and the discretised models is given in Figure 5.1 at the end of this section.

5.1.1 Geometric Design

Geometric design was done in the meshing software Gmsh [104]. Gmsh combines CAD functionality and 3D mesh generation capabilities. Geometries will be built based on a script in the Gmsh scripting language which defines geometrical entities, physical regions and options for the discretisation process. Gmsh also offers a graphical user interface. There are features that are only available in the scripting language therefore this approach was used in this thesis.

Geometric design consisted of two different types of work. One the one hand it meant to do geometric (re–)design of elements that are used in the real world experiments. Section 5.1.1.1 describes how this process is done in Gmsh. On the other hand I incorporated the segmentations obtained from medical image data as described in Chapter 4. During the course of this thesis I used two different approaches to handle this as will be described in Section 5.1.2.

5.1.1.1 CAD design

Gmsh offers two kernels to design the geometry, the built-in kernel and the open cascade kernel. While the open cascade kernel allows to shortcut design because some geometries (e.g. a box or a sphere) are predefined, the built-in kernel had to be used as I used features only implemented in the latter. Using the built-in kernel, the geometrical model has to be built by hand, starting defining the points of the features used, connecting the points to lines, defining surfaces by the lines enclosing them and finally defining volumes in the geometry by their boundary surfaces. In Gmsh, points, lines, surfaces and volumes are called *geometry entities* and they are identified by an integer. Geometrical entities can be grouped in so called *physical groups*. The physical groups are identified by an integer tag. The tag will be stored alongside the discretised data. This allows marking of different subregions and transferring this information to programs performing the numerical experiments.

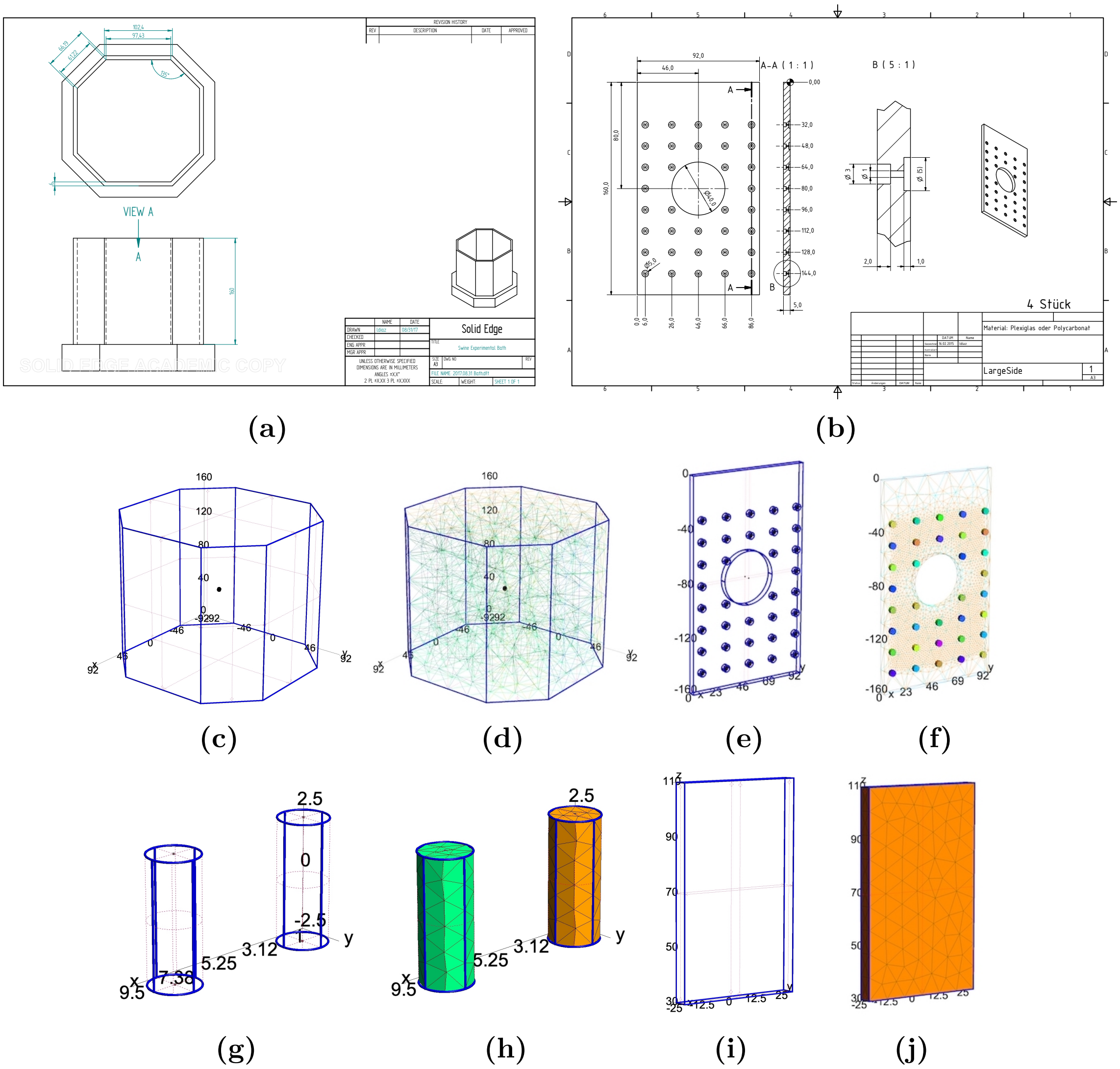
In order to produce flexible and reusable code, Gmsh allows to use *macros*. Macros resemble functions in other programming languages and can be reused in the geometrical design process. However macros do not run in their one scope but execute the macro as if the code was copied to the position where the macro was called. This means care had to be taken to not overwrite already used variables. A list of the macros interfaced for other users is given in 5.1. It is crucial to store the boundaries of the created entities to ensure that the entity can be separated from the surrounding volumes.

The physical tags associated with each region are written out on region creation alongside a string describing the region for automated reuse in the numerical experiments and for documentation, i.e. using the tag file specific regions can be identified by a name and not by remembering the right tag from the physical region definition.

Using this, a model for the experimental setup and for the multi-electrode measuring panels used in the experiment were built based on technical drawings provided by Laura Díaz. In addition a model for a pair of cylindrical electrodes to simulate a dipol and a model for shock electrodes based on the real experimental geometries were built. These were not constructed following technical drawings.

5.1.1.2 Including existing surface meshes

The segmentation process, c.f. Chapter 4 resulted in .stl surface mesh data. Gmsh allows to import these surfaces. Without further processing, Gmsh uses the surface discretisation and constructs the 3D mesh based on the already existing surface discretisation in the .stl file. Alternatively, a geometrical model will be built based on the discretised .stl surfaces and used to remesh the surface in Gmsh, see [111]. Both approaches have been used in this thesis. The latter proved more useful because it allows to skip the remeshing steps described in Section 4.2.1 and allows to use the Gmsh framework to set mesh densities and use the Gmsh algorithms to improve the



(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Figure 5.1 – *CAD-designed components used in the thesis. All design has been done in Gmsh. Subfigures 5.1a, 5.1a: technical drawings provided by Laura Díaz of the Langendorff perfusion bath and the multi-ECG panel. Subfigures 5.1d – 5.1f: corresponding CAD-models in Gmsh and spatially discretisations, respectively. Subfigures 5.1g — 5.1j: the CAD models of the dipol and the shock electrode and their spatial discretisation. These models were not based on a technical drawing.*

Table 5.1 – *CAD-design macros written in Gmsh, see listings in Appendix A.1.*

Macro name	Purpose	Input parameter	Output parameter
CreateBath	Create a model of the Langendorff perfusion bath	Height of the bath	Boundary surfaces of the volume
CreateDipol	Create a pair of cylindrical dipol electrodes	Distance between the cylinders, centre between the cylinders, radius of the cylinders, height of the cylinders, normal vector on the cylinder disc	Boundary surfaces of the electrodes
electrodepanel	Create a multi-ECG electrode panel	—	List of electrode volumes, boundary surfaces of the panel
electrode	Create a box	Position of one corner in space, length of the sides, normal and thickness of the box	Boundary surfaces of the box

mesh quality, see 5.1.2. If the surface meshes are imported without remeshing in Gmsh, the mesh has to be properly preprocessed, otherwise the solver solving the finite element problem does not converge [106]. Therefore either the imported mesh has to be of good quality while the mesh density will determine the density of the volumetric mesh close to the surface or it has to be remeshed in Gmsh.

Importing existing meshes

Importing a surface mesh in Gmsh is done by the code

```
Include "path/to/stl-file1.stl";
Include "path/to/stl-file2.stl";
```

at the appropriate position in the Gmsh code. The surfaces will be identified by the current surface counters which can be used to define the volume and the volume as hollow in the surrounding geometry. The mesh will not be changed in the 3D meshing process, i.e. a bad mesh quality will be passed on to the finite element mesh and hinder the solver. Therefore post-processing the segmentations was necessary and remeshing in the post-processing process every time the mesh density should be changed.

Remeshing in Gmsh

An alternative to post-processing the surface mesh is to remesh in Gmsh. To do so, after importing the surface mesh or surface meshes Gmsh classifies them based on the angle between the surface normal vectors, see [112–114], to define geometrical features that have to be kept in the remeshing process. Based on the classification, new geometrical models are created (in contrast to a discretised model, i.e. a mesh) which are used for the remeshing process. The downside is that referencing the surfaces in automated scripts thereafter is not possible as the surface numbering changes. To deal with this problem, I wrote an algorithm which groups connected surfaces and stores their identifying numbers. Together with the bounding box of the surfaces, this was used to detect the outer boundary of the heart and the surfaces therein, defining the heart chambers and make the surface identifiers accessible. The algorithm is attached in Listing A.4.

5.1.2 Spatial discretisation

Discretisation of the geometrical model defined in Section 5.1.1.1 was done in Gmsh using the frontal delaunay meshing algorithm in 2D and the delaunay meshing algorithm in 3D. The size of the discretisation cells can be either given by the minimum / maximum element size and done by Gmsh automatically or by using so called *mesh size fields* which provide a variety of options to setting the mesh size in the associated region(s). The initial discretisation of the geometry to a mesh thereafter can be optimised by Gmsh to improve the mesh quality. I made use of the built-in optimisation algorithm and the netgen [115] optimisation algorithm interfaced by Gmsh. In order to provide

Table 5.2 – *Gmsh fields and their usage.*

Field name	Purpose
Distance	Returns the distance to entities provided to the field
Threshold	This field takes as input a distance field, a minimal distance `DistMin` a maximal distance `DistMax`, a minimal value `LcMin` and a maximal value `LcMax`. Values from 0 to `DistMin` are set to `LcMin`, values greater than `DistMax` are set to `LcMax` and values between are interpolated linearly
Min	Returns the minimum of several input fields

'good enough' meshes I used the built-in optimisation several times thereafter the netgen mesh quality optimiser and the Gmsh built-in optimiser again.

Mesh size control with background fields

Together with Simon Wassing mesh size fields were implemented to control the mesh size. A mesh size field is a function on the spatial coordinates and returns a float value for the position in space. Several fields can be combined to a new field to make use of the various field generation mechanisms. The cell size of the discretisation will be calculated by Gmsh based on a single background field which has to be set. Further explanation is given in the Gmsh documentation. Table 5.2 gives an overview over the fields used in course of this thesis.

Variation of spatial discretisation mesh size is a common tool in finite element analysis to ensure the results are independent of the mesh size and therefore to exclude one source of errors [116, 117].

5.1.3 Numerical experiment setups

The geometrical elements described in the last section were combined to form numerical experiment setups. The corresponding setups and variations within these setups are presented in this section. The corresponding setups can be created by Gmsh scripts. Input parameters determine the discretisation mesh size and experiment specific parameters.

Defibrillation experiments setups

5.1.3.1 Langendorff experiments setups

The MRI segmentation described in Section 4.1.2 of the heart and two shock electrodes were put into the Langendorff model of the experimental setup. Additionally a rotation of the heart around the z-axis and a size factor were implemented using Gmshs mesh processing tools. The rotation angle and the size factor can be given to the script via the command line allowing automation. A screenshot of the setup before discretisation is shown in Figure 5.2a.

5.1.3.2 In-vivo experiments setups

The geometrical data obtained as described in Section 4.1.1 were surrounded by a cylinder mimicking the boundary of the animal with surrounding air. The heart segmented in the C-arm-data was replaced by the heart obtained from the MRI scan to have a suitable geometric model of the heart. Each position of the ICD in the medical image data was used for a geometrical model, i.e. the three ICD positions in the medical image data formed three geometrical models. The final geometric models included the heart model, the spine and the rips, one of the ICD positions and the catheter in the heart. A visualisation for one ICD position is shown in Figure 5.2b.

Multi-ECG experiments setups

One set of simulation consisted of the multi-ECG experiment setup with the Langendorff bath and the four multi-ECG electrode panels placed in the bath. Depending on the numerical experiment other geometrical models were placed in the bath in addition.

5.1.3.3 Langendorff bath experiments with dipol setups

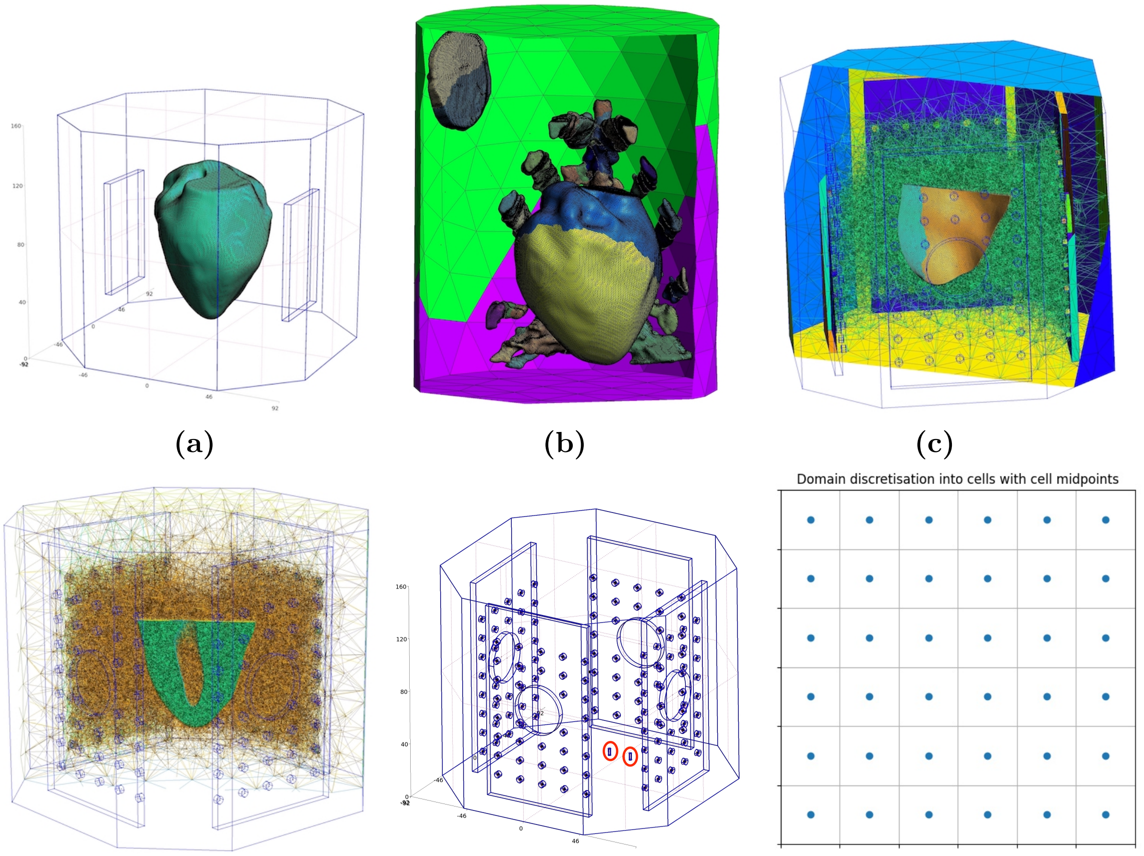
Four multi-ECG panels were placed into the Langendorff bath as well as a two cylindrical dipol electrodes. The position and orientation of the dipol can be varied by input parameters to the mesh as well as the radius and the height of the electrodes. The setup is shown in Figure 5.2e. Due to its small size, the dipol is difficult to spot. These data were not further used in this thesis.

5.1.3.4 Langendorff bath experiments with electrical cardiac activity setups

In order to simulate cardiac dynamics, the diffusion-MRI segmentation of the heart was placed in the Langendorff bath alongside four multi ECG panels. I implemented two versions of this setup. The first ignores the heart tissue and only discretising the heart surface, the bath and the multi-ECG panels, the second one includes the heart tissue and chambers into the discretisation process. Screenshots of the discretised experimental setups are shown in Figure 5.2d and Figure 5.2c, respectively. The 3D dynamics simulation were used in the lab rotation report of Master student Roland Stenger [118]. The surface tissue data were used in Section 7.2.2.

5.1.3.5 Abstract geometrical model setups

Another way to perform research on cardiac dynamics is to prescind from the geometrical model and simulate a 2D rectangular region. Flat 2D surfaces have been (implicitly) discretised into regular grids. An example is given in Figure 5.2f.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 5.2 – *Overview over the finite element setups. Figures 5.2a – 5.2b show the current density simulation setups in Langendorff perfusion and in-vivo. The Langendorff perfusion setup had implemented rotation of the heart around the z-axis and scaling the heart using Gmsh. The setup is shown pre-meshing, consequently only the imported surface mesh is already discretised. Figure 5.2c – 5.2d show the numerical setups used for cardiac dynamics simulations. During preprocessing, the heart the atria is removed from the heart model such that only dynamics on the ventricles is simulated. This setups was implemented using two approaches, first to simulate the dynamics only on the surface of the heart (to reduce computation time as the simulation data of interest were the ECG signals that only depend on the surface data), Figure 5.2c and also including the heart muscle, Figure 5.2d. As these setups are shown after discretisation, it can be seen how either the interior of the heart is discretised or not. In both setups the multi-electrode panels can be identified by their CAD-representation. Lastly Figure 5.2e shows a numerical setup that has been used to reconstruct the position of a dipol within the bath. The dipol can be spotted in the lower part of the bath. Figure 5.2f shows an experimental model where the domain is discretised in a regular grid. Usually variables are defined on the cell centres which then represents the corresponding value of the whole cell.*

5.2 Numerical Experiments

The spatial models introduced in the last section were used to conduct different types of numerical experiments. In this section's first subsection I will introduce experiments conducted to better understand the distribution of current in defibrillation experiments The second subsection will introduce the experiments conducted to model electrical activity in and on the heart both, within the Langendorff- setup and within a very general model of heart surface tissue.

5.2.1 Current density experiments

The current density experiments simulate the current density distribution within the experimental setups. To solve the mathematical equations, the finite element method was used. The corresponding mathematical equations used were passive volume conductor equations following the monograph [69] and neglect electrochemical and other effects. The corresponding equation for a domain Ω is

$$\nabla\sigma\nabla\phi = 0, \tag{5.1}$$

with σ being possibly inhomogeneous conductivity and ϕ being the electrical potential. These experiments include Dirichlet boundary conditions to mimic an electrical shock with a certain potential difference and Neumann boundary conditions elsewhere. Those complete the system,

$$\begin{aligned} \phi &= \phi_{b,1} && \text{on} \quad \partial\Omega_{D,1}, && (5.2)\\ \phi &= \phi_{b,2} && \text{on} \quad \partial\Omega_{D,2}, && (5.3)\\ \nabla\phi\cdot\vec{n} &= 0 && \text{on} \quad \partial\Omega\setminus\Omega_{D,1,2}, && (5.4) \end{aligned}$$

where $\phi_{b,1}$ and $\phi_{b,2}$ are the values at the boundaries $\partial\Omega_{D,1}$ and $\partial\Omega_{D,2}$ respectively, $\vec{n}$ is the normal vector at the boundary and $\partial\Omega$ represents the boundary. In order to apply the finite element method, this system of equations has to be transformed into the weak formulation as mentioned in Subsection 2.6.2. The system of Equations 5.1 and 5.4 becomes

$$\int_\Omega \mathrm{d}x\sigma\nabla\phi\nabla v = 0 \qquad \forall v\in V, \tag{5.5}$$

with V being the corresponding function space. This mathematical model was used in the following setups:

- Setup dipol in bath: the Dirichlet boundary conditions are used to apply a potential difference between the two dipol electrodes, the ECG-electrodes measure the corresponding field, c.f. Section 5.1.3.3.

- Setup in-vivo: pig bone structure of the torso, a heart and two electrodes are placed in a cylindrical model of a pig body. The current density distribution is investigated, c.f. Section 5.1.3.2.
- Setup ex-vivo: A heart model and two shock electrode models are placed in the Langendorff perfusion bath model. Both, homogeneous and inhomogeneous conductivity for the heart were used, c.f. Section 5.1.3.1.

5.2.2 Continuous current

With the discretisation of the Equation 5.5, the potential ϕ will not be differentiable. Therefore in order to calculate the current according to Ohm's law [119]

$$\vec{\jmath} = -\sigma \nabla \phi,$$

an alternative approach was used. The current $\vec{\jmath}$ is explicitly solved for by introducing this variable in the Poisson equation, Equation 5.1, resulting in

$$\nabla \vec{\jmath} = 0.$$

This equation will be solved in addition the Poisson problem. So in total, the problem to be solved is

$$\begin{aligned}
\nabla \vec{\jmath} &= 0 \qquad \text{in}\,\Omega,\\
\vec{\jmath} + \sigma \nabla \phi &= 0 \qquad \text{in}\,\Omega,\\
\vec{n}\vec{\jmath} &= 0 \qquad \text{in}\,\partial\Omega,\\
\phi &= \phi_o \qquad \text{in}\,\partial\Omega,
\end{aligned}$$

where the second equation relates the current to the potential and allows to apply the Dirichlet boundary conditions for the potential ϕ.

The variational form for this equation needs a function space for each variable. Let Σ be the vector function space for $\vec{\jmath}$ and let V be the function space for ϕ as before. Let $\vec{\tau} \in \Sigma$ and $\psi \in V$ be the corresponding test functions then the variational form becomes

$$\int_\Omega \mathrm{d}x \vec{\jmath}\vec{\tau} - \phi \nabla \vec{\tau} + \int_{\partial\Omega} \phi \vec{\tau}\vec{n} \mathrm{d}s = 0 \qquad \forall \vec{\tau} \in \Sigma, \tag{5.6a}$$

$$\int_\Omega \mathrm{d}x \psi \nabla \vec{\jmath} = 0 \qquad \forall \psi \in V. \tag{5.6b}$$

5.2.3 Electrical Dynamics Experiments

Electrical dynamics experiments were conducted using the monodomain equation described in Section 2.5. This was followed by a reconstruction of the extracellular potential following the approach described in [120]. Thereafter a diffusion equation was solved to diffuse this potential into the surrounding bath using the approach described in Section 5.2.1 with the potential at the heart surface as Dirichlet boundary condition. The monodomain equation from Equation 2.7 was

$$\frac{\lambda}{1+\lambda}\nabla(\sigma_i \nabla v_m) = \xi C_m \frac{\partial v_m}{\partial t} + \xi I_{\text{local}}.$$

Together with this equation, usually no-flux boundary conditions are used, i.e.

$$\nabla\phi \cdot \vec{n} = 0 \qquad \text{on} \qquad \partial\Omega.$$

for the domain Ω.

5.2.3.1 Realistic geometry experiments

Realistic geometry experiments were conducted using the finite element method. In order to transform the monodomain equation into its variational form it was multiplied with a test function w and integration by parts, c.f. Section 2.6. This leads to

$$\int_\Omega \mathrm{d}x \frac{\lambda}{1+\lambda}(\sigma_i \nabla v_m)\nabla w = \int_\Omega \mathrm{d}x \left(\xi C_m \frac{\partial v_m}{\partial t} + \xi I_{\text{local}}\right) w. \tag{5.7}$$

Temporal discretisation was done using the finite difference approach, i.e. the temporal domain is split into N intervals of (fixed) size Δt. This means a time interval $[t_0, T]$ is discretised into intervals of size Δt, the discrete times are

$$t_i = t_0 + i\Delta t.$$

Time stepping was done using the forward Euler method [121]. That means, the time derivative of $\frac{\partial v_m}{\partial t}$ is approximated by

$$\frac{\partial v_m}{\partial t} \approx \frac{v_m^{n+1} - v_m^n}{\Delta t}.$$

where the superscripts n and $n+1$ denote the current time step and the next time step, respectively, and Δt is the time discretisation size.

Applying this discretisation, the variational form becomes

$$\int_\Omega \mathrm{d}x \frac{\lambda}{1+\lambda}(\sigma_i \nabla v_m^n)\nabla w = \int_\Omega \mathrm{d}x \left(\xi C_m \frac{v_m^{n+1} - v_m^n}{\Delta t} + \xi I_{\mathrm{local}} \right) w. \tag{5.8}$$

In the numerical experiments, the heart was not isolated but placed into a conductive bath. The approach to model the electrical dynamics with the monodomain equation in an isolated environment and then to recover the extracellular potential ϕ_e from the bidomain equations was applied before [120, 122].

Based on the transmembrane voltage v_m^n at time $n\Delta t$, the extracellular potential is calculated following [120] by solving

$$-\nabla(\sigma_i + \sigma_e)\nabla\phi_e^n = \nabla\sigma_i\nabla v_m^n,$$

where σ_i and σ_e are the intracellular and extracellular conductivity tensors.

The corresponding variational formulation again is obtained by multiplication with a test function w and integrating by parts. Together with no-flux boundary conditions, this results in

$$-\int_\Omega \mathrm{d}x(\sigma_i + \sigma_e)\nabla\phi_e^n\nabla w = \int_\Omega \mathrm{d}x\sigma_i\nabla v_m^n\nabla w.$$

The solution of this equation is only defined up to a constant. Physically this makes sense as the quantity reconstructed is a potential without any reference given. This has to be taken into account when solving the linear system that corresponds to this problem.

Lastly, the reconstructed potential at the interface of the heart with surrounding tissue was taken as Dirichlet boundary condition to a Laplace equation to simulate the diffusion process into the vicinity. The corresponding mathematical framework is similar to the one described in Section 5.2.1.

5.2.3.2 Abstract geometry experiments

Using the abstract geometrical model, the monodomain approximation of cardiac dynamics has been employed. Rearranging Equation 2.7 gives

$$\frac{\partial v_m}{\partial t} = \frac{1}{\xi C_m}\frac{\lambda}{1+\lambda}\nabla(\sigma_i\nabla v_m) - \xi I_{\mathrm{local}}.$$

The discretisation in time was done using the explicit Runge-Kutta method of order 5(4) [123] from scipy. Discretisation of the Laplace operator was done using the 5-point stencil, c.f. Section 2.6.4.

5.2.3.3 Local current

The monodomain equation has one term that hasn't been treated yet, the current through the cell membrane, I_{local}. Equation 2.3 in Section 2.5 relates the single cell dynamics to the term used in the spatial model. Equation 2.2 gives the general form of the transmembrane current,

$$\frac{\mathrm{d}v}{\mathrm{d}t}_{\text{local}} = f(v, \vec{w})$$
$$\frac{\mathrm{d}\vec{w}}{\mathrm{d}t} = g(v, \vec{w}),$$

where v is the transmembrane voltage, $\vec{w}$ is a vector of state variables and f and g depend on the model used. Throughout this thesis I have used the Mitchell-Schaeffer model which has the following form:

$$\frac{dv}{dt} = \frac{wv^2(1-v)}{\tau_{\text{in}}} - \frac{v}{\tau_{\text{out}}} + J_{\text{stim}}(t) \tag{5.9}$$

$$\frac{dw}{dt} = \begin{cases} \frac{1-w}{\tau_{\text{open}}}, & \text{if } v < v_{\text{gate}} \\ \frac{-w}{\tau_{\text{close}}}, & \text{if } v > v_{\text{gate}} \end{cases} \tag{5.10}$$

with $J_{\text{stim}}(t)$ as stimulus current, injected to the system from outside. The parameter v_{gate} models the opening and closing of ionic channels and the parameters τ_{in}, τ_{out}, τ_{close}, τ_{open} can be related to the initiation, the decay, the plateau and the recovery phase of a cardiac action potential, respectively. The parameter values are given in Tabular 5.3. The corresponding action potential is shown in Figure 5.3.

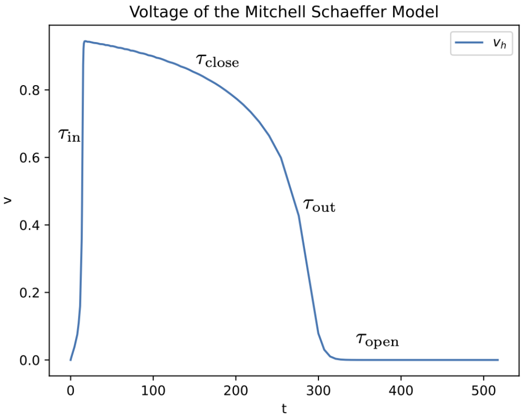


Figure 5.3 – *Example time series of the local Mitchell-Schaeffer voltage in zero spatial dimensions (no diffusion). The time constants of the model can be approximately identified with the phases of the action potential.*

Table 5.3 – *Mitchell Schaeffer simulation parameters 2D.*

v_{gate}	τ_{in}	τ_{out}	τ_{close}	τ_{open}	σ
0.13	0.3	6	120	150	0.001

Table 5.4 – *Solver and preconditioner methods of the experiments used by dolfin.*

Experiment	preconditioning method	solving method
Current density	Algrebaric multigrid	Generalized Minimal Residual
Cardiac dynamics: Monodomain	petsc_amg	cg
Cardiac dynamics: Potential recovery	ilu	Generalized Minimal Residual
Cardiac dynamics: Diffusion	hypre amg	cg

5.2.3.4 Spatial discretisation

The solutions of the variational formulations in Sections 5.2.1 – 5.2.3 were discretised using the finite element method with continuous piecewise linear basis functions, i.e. using the Lagrange element [74, p. 72] for the realistic geometries. In the case of the approach used to calculate continuous current presented in Section 5.2.2, the current $\vec{\jmath}$ was approximated piecewise linear elements while the potential ϕ was approximated using piecewise constant functions for stability reasons [124]. The solver for the continuous current was implemented by Simon Wassing based on my initiative. The corresponding systems of linear equations were assembled based on the variational formulations with the python package dolfin [125–128] and solved within this package.

The abstract geometry were spatially discretised with the five point stencil, c.f. Section 2.6.4 turning the PDE into a set of ordinary differential equations.

Solvers for the FEM related system of linear equations

The dolfin package is built upon the PETSc library [129] and has access to the solver methods implemented therein. The solver used in this thesis were Krylov solvers using the methods indicated in Table 5.4.

Chapter 6

Current density in-vivo and ex-vivo

In the research group biomedical physics experiments have been conducted in Langendorff perfusion and in-vivo. To understand these experiments better, Simon Wassing, a master student supervised by me, and I performed a number of numerical experiments aiming at answering the following questions in Langendorff experiments:

1. How much current in a shock is *actually* applied to the heart in a defibrillation shock experiment?
   (a) How do the size and the rotation of the heart relative to the electrodes affect the heart?
   (b) How does the position of the heart relative to the electrodes affect the current penetrating the heart?
2. How do simulations with isotropic conductivity differ from simulations incorporating fibre orientation in the heart muscle?

The results presented here were also used in the Master's thesis of Simon Wassing [98]. Simon implemented and calibrated the actual numerical experiments. My part consisted in developing the numerical framework and finding, learning, testing and teaching the tools we used.

Furthermore, I performed first simulations to investigate the effect of the position of an implantable cardioverter-defibrillator (ICD) on the defibrillation current. Dr. Claudia Richter provided medical image data of a Göttingen minipig with three ICDs implanted as was shown in Chapter 4, Figure 4.3. The corresponding research question can be formulated as follows:

1. How does the position of the ICD in the body affect the defibrillation current?

Table 6.1 – *Simulation parameters.*

Parameter	Heart position experiment (isotropic)	Comparison isotropic / anisotropic conductivities
σ_h [µS/cm]	3731 [130]	900
σ_t [µS/cm] [98]	17700	17700
σ_l [µS/cm] [69]	—	3000
σ_t [µS/cm] [69]	—	1000
σ_n [µS/cm] [69]	—	315.25

6.1 Langendorff perfusion

6.1.1 Size and rotation numerical experiments

The size and rotation experiments were conducted by Simon Wassing including finding the right meshing parameters. The parameters used are indicated in Table 6.1. As voltage difference $\Delta\phi = 40\,\mathrm{V}$ was used. The conductivity of the Tyrode solution used in the group's experiments was measured by Simon Wassing [98]. The conductivity throughout the heart was assumed to be isotropic. The heart was either rotated with the septum parallel to the electrodes or — rotated by 90° around the z-axis — with the septum perpendicular to the electrodes.

We scaled the size of the heart by a factor α while the heart was kept centralised in the centre of the bath between the electrodes. As described in Section 5.1.3.1 I have implemented the rotation and scaling of the heart. We scanned the range $\alpha \in [0.7, 1.2]$ where the maximum α was restricted by the size of the bath. We did this to investigate the effect of the heart size on the current density for one constant voltage difference applied. This is important for two reasons: the heart size increases during the Langendorff experiments because it soaks itself with the Tyrode solution (this was quantified in the thesis of Simon Wassing [98] via the weight of the heart before and after the experiment) and this is important to compare the hearts of different animals (which have individual heart sizes), too.

To answer the introductory questions at the beginning of this chapter, we used the continuous current approach and calculated the amount of current through the heart I_h and through the Tyrode I_Tyrode and the total current I_total:

$$I_\mathrm{total} = \int_{\partial\mathrm{electrode}} \mathrm{d}s\vec{n}\vec{\jmath}, \tag{6.1a}$$

$$I_\mathrm{h} = \frac{1}{2}\int_{\partial\mathrm{H}} \mathrm{d}s\sqrt{(\vec{n}\vec{\jmath})^2}, \tag{6.1b}$$

$$I_\mathrm{Tyrode} = I_\mathrm{total} - I_\mathrm{h}, \tag{6.1c}$$

where $\vec{n}$ is the surface normal vector, $\vec{\jmath}$ is the current density, ∂electrode the surface of one of the electrodes and ∂Heart the surface of the heart.

Figure 6.1 summarises the results. It can be seen that the rotation does not play a role in these simulations. The amount of current through the heart is 7 % for the scale factor $\alpha = 0.7$ and goes up to 23 % for the scale factor $\alpha = 1.2$. The total current decreases in that range from 8.85 A to 7.87 A. The current through the heart scaled with the volume of two the power of 2/3, $I_\mathrm{h}/V_\mathrm{h}^{2/3}$ is approximately constant. It varies over the range by about 10 %.

6.1.2 The effect of translation

To answer how translating the heart with respect to the centralised position between the electrodes changes the current, the heart was translated in the normal direction to the electrodes (x-direction) by $\Delta x = \pm 15\,\mathrm{mm}$, parallel to the short side of the electrodes (y-direction) in from $-40\,\mathrm{mm}$ to $40\,\mathrm{mm}$ in steps of $\Delta y = 10\,\mathrm{mm}$ and in height (z-direction) from $-10\,\mathrm{mm}$ to $30\,\mathrm{mm}$ in steps of $\Delta z = 5\,\mathrm{mm}$.

Figure 6.2 shows the current penetrating the heart in fraction of the maximum current calculated in these numerical experiments to compare the results more easily. The highest current was calculated for $\Delta x = -15\,\mathrm{mm}$, $\Delta y = 0\,\mathrm{mm}$, $\Delta z = -20\,\mathrm{mm}$. Increasing the height reduced the current as well as moving the heart in y-direction from the centre ($\Delta y \neq 0\,\mathrm{mm}$). Moving the heart towards the electrodes in x-direction increased the current through the heart.

6.1.3 The effect of anisotropy

To investigate the effect caused by the simplification of the conductivities from anisotropic to isotropic, Simon Wassing performed numerical experiments to find an equivalent isotropic conductivity σ_h for the given fibre, sheet and sheet-normal conductivities (c.f. Table 6.1). The resulting conductivity value was found by performing a set of numerical simulations and calculated the current through the heart. Based on these data a fit through the data was done of the form $I_\mathrm{h}(\sigma_h) = A \cdot \sigma_\mathrm{h}^b$, c.f. Figure 6.3a, and by inverting this function $\sigma_h = 900\,\mathrm{\mu S/cm}$ was found.

To compare the results of the simulation with and without isotropic conductivities the current density distributions within the heart were compared. Figure 6.3b shows the reversed cumulative current density distribution for the isotropic and the anisotropic case. It can be seen that the distribution of current densities in the anisotropic case is broader. Also higher current densities appear more often in the anisotropic case.

6.1.4 Discussion

We found that the current that is applied to the heart depends on the position of the heart relative to the shock electrodes. Depending on the position within the electrodes we found a variance of up to 20 % of the maximum current when moving the heart out of the centred position between the electrodes. What does this mean for experiments? In experiments that research defibrillation care has to be taken to the electrode positioning. Based on these results moving the heart or one or both electrodes during an experiment hampers comparability between shock strength needed

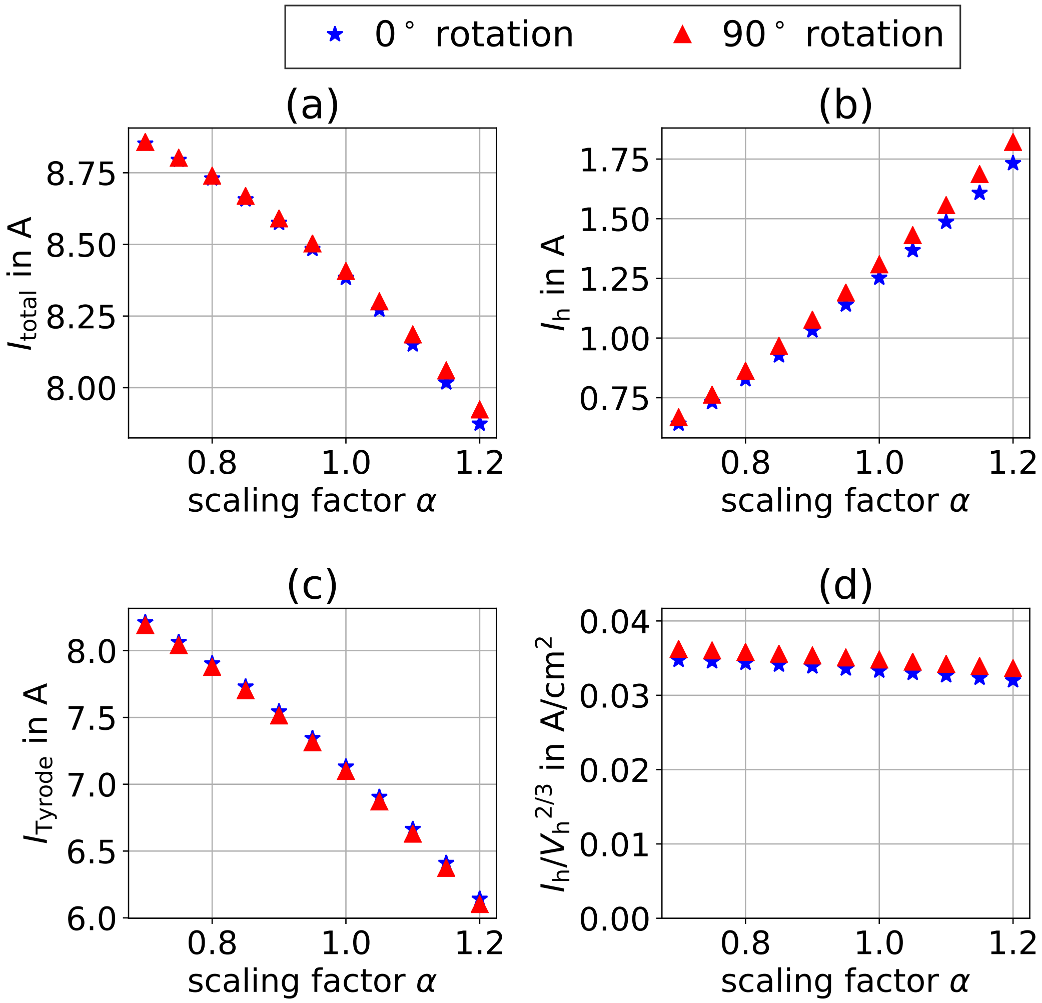


Figure 6.1 – *The currents in the heart during Langendorff defibrillation resizing the original heart size by the factor α with isotropic conductivity. The total current (Figure 6.1 (a)), the current through the heart (Figure 6.1 (b)), the current through the Tyrode solution (Figure 6.1 (c)) and the total current scaled by the heart volume (Figure 6.1 (d)). Note that rotating the heart by 90° around the z-axis does hardly change the currents. The rotation 0° refers to the septum being parallel with the electrodes, the angle 90° refers to the septum being perpendicular to the shock electrodes. Note that due to the high difference in conductivities from the Tyrode solution and the heart (c.f. Table 6.1) a large amount of the current passes the heart through the nutrient solution. Interestingly Figure 6.1 (d) shows that the current through the heart I_{heart} reciprocally scaled by the heart volume to the power of $2/3$, $V^{2/3}$ is constant.*

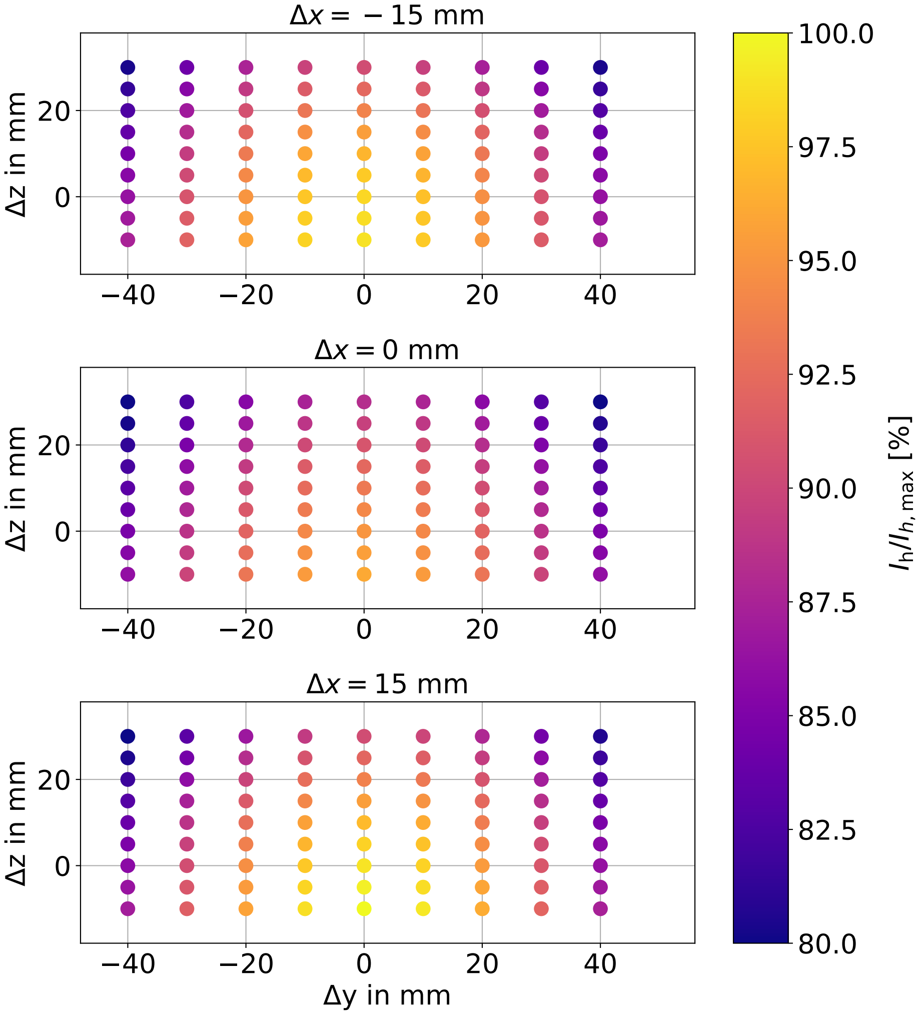
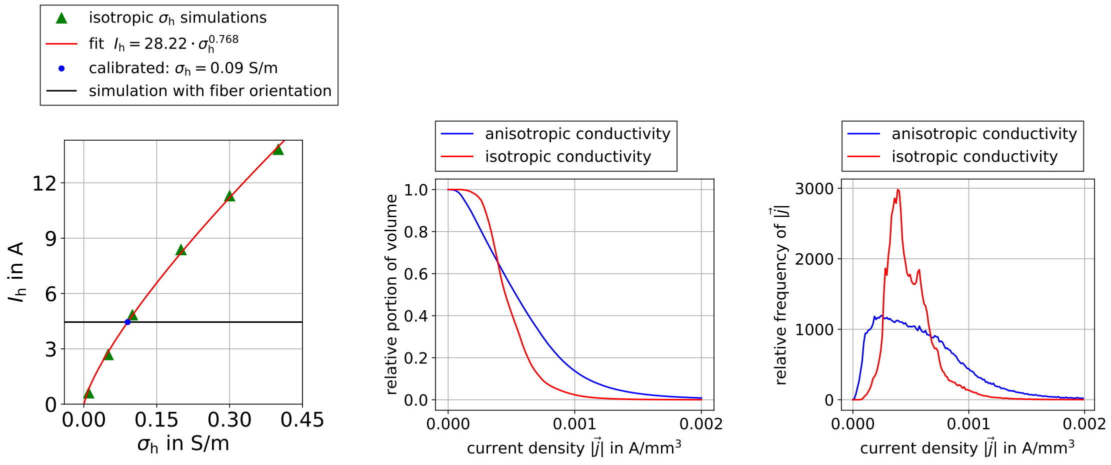


Figure 6.2 – *The current through the heart I_h normalised with respect to $I_{h,\max}$ when moving the heart from the centred position between the electrodes ($\Delta x = \Delta y = \Delta z = 0$) towards the electrodes (Δx changes), higher or lower (Δz changes) or parallel to the short side of the electrodes (Δy changes). The measurements of the current through the heart are normalised by the maximum current. This was at the position $\Delta x = -15$, $\Delta y = 0$, $\Delta z = -20$, $I_h/I_{h\,\max}$. It can be seen that the current through the heart is symmetrical in x-direction. Lowering the heart position increases the current penetrating the heart ($\Delta z < 0$) while moving the heart parallel to the short side of the electrodes out of the centre. These simulations have been performed with isotropic conductivities.*



(a) **(b)** **(c)**

Figure 6.3 – *Results from current density distributions. Figure 6.3a: calibration of the isotropic simulations. Heart currents I_h in the isotropic simulation to find the (isotropic) conductivity equivalent σ_h to obtained the same current through the heart I_h as in the anisotropic simulations. The latter is indicated as black line in the plot. A function of the form $I_h(\sigma_\mathrm{h}) = A \cdot \sigma_\mathrm{h}^b$ was fit to the data to obtain $\sigma_\mathrm{h} = 0.9\,\mathrm{S/m} = 900\,\mathrm{\mu S/cm}$. Figure 6.3b shows the reversed cumulative density distributions for both cases. It can be seen that the distribution of values within the anisotropic case is broader compared to the isotropic case. Figure 6.3c highlights this. While the majority of the isotropic values are in the range of $0.25\,\mathrm{mA/m^3}$ to $0.75\,\mathrm{mA/m^3}$, including anisotropic conductivity broadens the ranges of current conductivity values in the heart tissue.*

for defibrillation before and after the repositioning. This effect can be explained by the difference in conductivity between the heart and the surrounding solution: As an effect of the difference in conductivities the current 'avoids' the heart to minimise the total resistance between the electrodes.

We found that the current interacting with the heart depends on the heart size: for the largest heart ($\alpha = 1.2$) we found that 15 % of the current penetrate the heart (1.75 A) while for the smallest heart ($\alpha = 0.6$) 8 % of the total current penetrates the heart (0.7 A). This means quantities that quantify a defibrillation shock, commonly e.g. the energy used, should be normalised to the heart used in order to be comparable with each other: For a given current applied in the same experimental setup, a small heart would interact with less current compared to a larger heart.

In addition we saw that the current through the heart scaled by the volume to the power of 2/3 ($I_\mathrm{h}/V^{2/3}$) remains approximately constant. This can be explained if we note that term $V^{2/3}$ resembles a surface. As we have scaled the heart volume by the factor α we scaled the surface by the factor $\alpha^{2/3}$. As the current through the heart is calculated by the integral over the surface, if the surface is smaller by this factor, the same is true for the current through the surface. The small decrease that can be seen in Figure 6.1 panel (d) is probably caused by a reduction of the total current due to the higher resistance of the experimental setup when the heart size is increased.

In the simulations we performed we could not find a considerable effect on the current applied to the heart when changing the rotation angle by 90°. However, in these simulations the fibre orientation was not taken into account. Including fibre orientations I would expected an effect. Figuratively speaking, if the electrical field is more aligned with the fibre direction the overall conductivity of the heart would be higher compared to if the electrical field is perpendicular to the fibre direction. This needs further investigation to quantify whether this effect is also negligible for the geometric model of the heart including fibre orientation.

Simulations with fibre orientation showed in Figure 6.3 a wider distribution of current densities within the heart muscle compared to isotropic conductivity simulations when both hat the same total current through the heart. This means in the heart there are more different current densities. As a next step it would be interesting to see what the effect of this is on defibrillation. This can be done by cardiac dynamics simulations where this current is incorporated as external stimulus.

Simon Wassing and I have neglected electrochemical effects in these in-sito studies. One way how electrochemical effects would change the results is quantitatively: double layers of ions form close to the electrodes and damp the effective voltage applied to the heart. Another way are electrolytic effects where due to the current applied chemical reactions happen. This would be seen in real experiments by non-linear changes of the resistance of the Tyrode. Another source of error might be induced by spatial discretisation error. However Simon Wassing conducted mesh convergence studies to show that the results are independent of the mesh size chosen. This does not exclude errors due to simplification in the geometrical modelling process. It would be very surprising if these effects change the qualitative results of this numerical studies.

Table 6.2 – *In-vivo currents.*

ICD-label	current [A]
ICD-old	0.61
ICD-new	0.71
SICD	0.73

6.2 In-vivo simulations

Medical data provided by Dr. Claudia Richter gave three different positions of an ICD relative to the heart in-vivo. The geometrical model was fully obtained from the medical image data as described in Section 4.1.1 and Section 5.1.3.2. The positions and the corresponding labelling is given in Figure 6.4a.

The conductivity of the heart was chosen as indicated in Table 6.1, the conductivity of the body and the volumes forming the hollows in the heart were chosen to the value of the Tyrode solution, c.f. Table 6.1 (see Discussion). The potentials at the electrodes was ±10 V, respectively. The bones were considered no-flux boundary conditions for the current because the conductivity of bones is several orders of magnitude smaller than the conductivity of muscle tissue and blood (conductivity of bones $\sigma_{\text{bone}} \approx 10^{-4}\,\mu\text{S/cm}$ [131]). The mathematical model employed was the mixed-domain formulation in order to obtain a continuous current (Section 5.2.2).

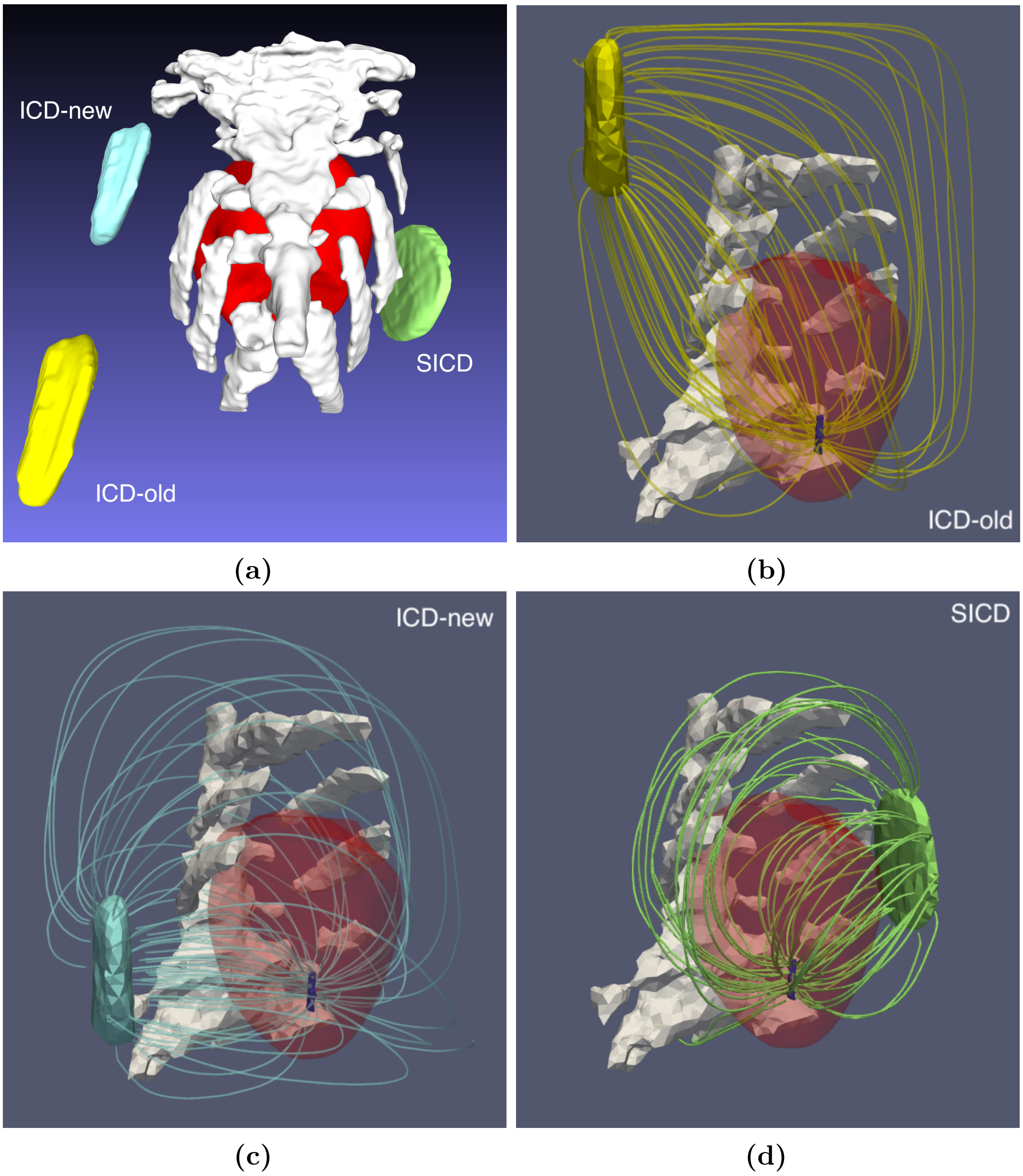
The current over the heart surface was calculated over the closed surface of the heart:

$$I_{\text{h}} = \int_{\partial\text{H}} \text{d}s\vec{n}\vec{\jmath}. \tag{6.2}$$

with heart surface ∂H, $\vec{n}$ the normal vector to the heart surface and $\vec{\jmath}$ the current density. In addition streamline plots were made using Paraview [99, 100, 132]. Table 6.2 shows the calculated heart currents I_{h}. They are in the same order of magnitude for the three positions. The total current through the heart in the old positioning is approximately 14 % smaller compared to the new position and the SICD position. Figure 6.5 shows the current density distributions for the three positions. The lowest current density inside the heart of the ICD-old position is higher compared to the other two positions indicated as the later rise of the distribution. The current distribution of the positions ICD-new and SICD resemble each other very much. The three current density distributions align approximately from $1 \times 10^{-4}\,\text{A/mm}^3$ on and are zero from approximately $2.5 \times 10^{-4}\,\text{A/mm}^3$ on.

6.2.1 Discussion

These simulations are very preliminary. For example the conductivity of the body tissue was set to be the one of Tyrode solution because the geometrical model used in Section 6.1 was changed but not the corresponding parameters. This is an effect of this numerical setup still being in development (and that Simon Wassing had left the group



(a) **(b)**

(c) **(d)**

Figure 6.4 – *Figure 6.4a: Labels of the positions of the ICDs in the in-vivo model simulations. The position ICD-old (yellow) was used in the group's in-vivo experiments until the position was changed to ICD-new (light blue). In addition an ICD was implanted to the position of subcutaneous ICDs (SICD, light green). The bone structures are indicated in white, the heart is indicated in red. Figure 6.4b — 6.4d: Streamlines through the heart for the different ICD positions.*

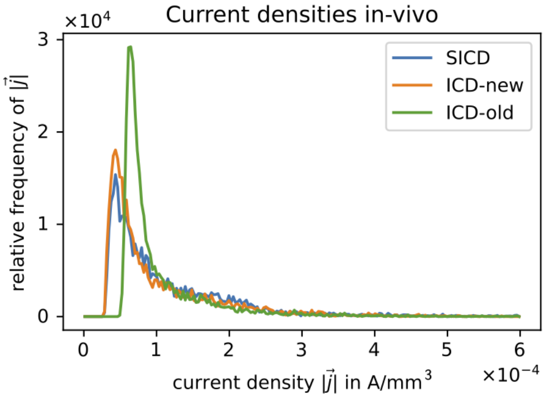


Figure 6.5 – *Current density distribution inside the heart for each ICD position indicated in Figure 6.4. The ICD-old position forms a sharper and higher peak compared to the ICD-new and the SICD positions. In addition it raises at higher current densities. The ICD-new and the SICD position have a comparably similar shapes. The peak of the SICD position is slightly higher.*

after his Master's thesis while I was focussing on the cardiac dynamic simulations). In literature the conductivity for tissue in the body is rather in the magnitude of cardiac tissue. For example $\sigma_{\text{body}} = 2390\,\mu\text{S/cm}$ was used [133] which is smaller than the conductive of the heart. This would mean that the current 'prefers' paths through the heart muscle compared to the parameter values I used in which the current prefers to leave the heart into the body.

As this simulation was done with the Tyrode solution it is too early to draw a conclusion for the real experiments.

Nevertheless this is only a problem of the choice of parameters. The numerical framework has been developed and proved to be usable. The next steps will be to redo this simulation with the conductivity value σ_{body} from above and to find a spatial discretisation fine enough that the results are independent of it using the recently finished mesh-density analysis workflow.

This next step can easily (and should) be combined with the fibre orientation data. I expect that in this set of numerical experiments this will have an effect as the electrode is rotated around the heart with the same argumentation used in Section 6.1.4.

I have finished the development of the numerical framework for this numerical experiment and I obtained first results with this framework. In addition the next steps are clear and if there is time it will be very interesting to see the results.

Chapter 7

New Concepts

7.1 A measure for excitable media distance

The inverse ECG research reconstructs the underlying source on the heart based on the ECG-signals. The results have to be validated, so they have to be compared to the real source. Naïvely this is done using the Euclidean distance, i.e. let a spatial domain be discretised into N pixels and let ϕ^i_{rec} be the reconstructed solution of the true underlying value ϕ^i_{orig} for the pixel with index i, then the difference between the reconstructed solution and the underlying source would be calculated as

$$d_{\text{E}}(\vec{\phi}_{\text{rec}}, \vec{\phi}_{\text{orig}}) = \sqrt{\sum_i \left(\phi_{\text{rec},i} - \phi_{\text{orig},i}\right)^2}. \tag{7.1}$$

This distance measure is for example used in the cost function in machine learning approaches to solve problems in excitable media research, e.g. in [134]. Another measure in use [135, 136] is the root mean squared error,

$$RMSE(\vec{\phi}_{\text{rec}}, \vec{\phi}_{\text{orig}}) = \sqrt{\frac{\sum_i \left(\phi_{\text{rec},i} - \phi_{\text{orig},i}\right)^2}{N}}, \tag{7.2}$$

which calculates the Euclidean distance between the voltage values of each pixel value and uses the mean value for the extended system. While these are well suited measures to compare the measured and reconstructed variables they fail in specific situations on the excitations cycle to properly describe the distance between the physical states.

This is exemplified using the 2-variable Mitchell Schaeffer model. Figure 7.1 shows the transmembrane voltage v time series after a short perturbation of the fixed point. After a steep rise it changes little for a rather long time (*refractory phase*) until it goes back to the resting v value. Table 7.1 gives the Euclidean distances, Equation 7.1 for the variable v between four points A, B, C, D before the rise, at the tip of the action potential, at (roughly) the first point in time of the refractory phase and at the last

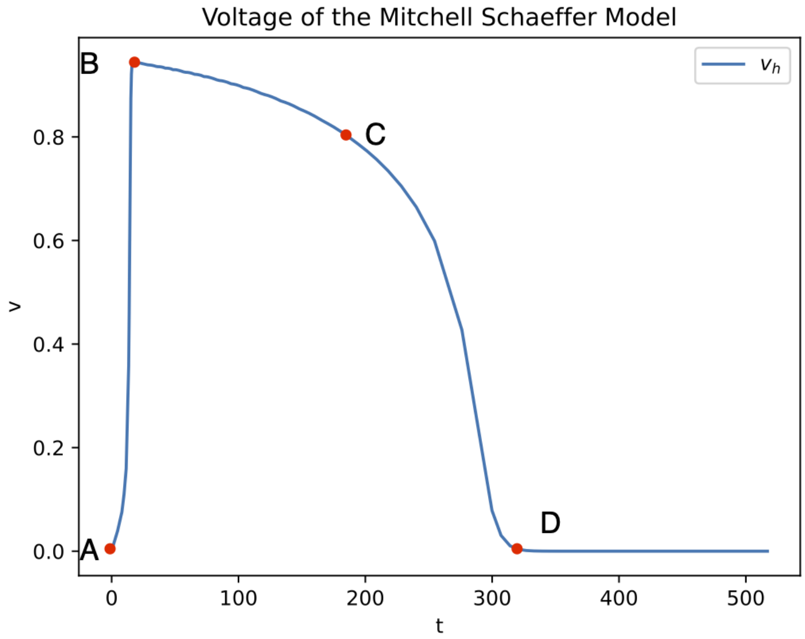


Figure 7.1 – *States that are very close in time are not close in state space. Figure 7.1 shows the time series of the voltage component calculated by the Mitchell Schaeffer model with parameters $\tau_{\text{close}} = 150$, $\tau_{\text{open}} = 120$, $\tau_{\text{out}} = 6$, $\tau_{\text{in}} = 0.3$, $v_{\text{gate}} = 0.13$, $j_{\text{stim}}(t) = 0.01$ for $t < 10$ and $j_{\text{stim}}(t) = 0$ for $t \geq 10$. Four points, A, B, C, D are marked to compare their distance in voltage. Point A and D are very close in this measure, however it takes a long time to reach from one state into each other. On the other side Point A and B are very close to each other in time however the distance in voltage is very high. This means the distance between voltage does not properly reflect how close the states are. Figure 7.2 explains this.*

point of that phase, respectively, are given in. It stands out that first, the difference of the transmembrane value v between points A and B is very high while the time to reach point B from A is very short. On the other hand, the distance in v between point B and C and B and D is rather similar, however the time it takes to come from B to C is significantly smaller compared to the time from B to D. The phase space trajectory corresponding to the time series depicted in Figure 7.2 underlines this. The time it takes for the transmembrane voltage variable to go back to its resting state value is only approximately 60 % of the time the physical state needs to go back into its fixed point.

An alternative way to measure the distance in state space based on the temporal development will be presented in the next section.

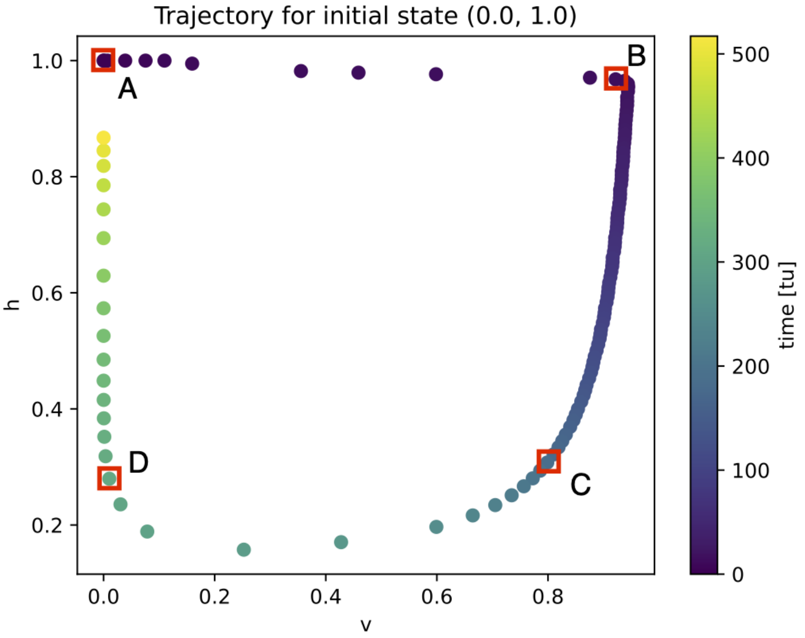


Figure 7.2 – *The trajectory in state space is indicated with the time from the resting state colour coded. Red squares indicate the positions of the points A, B, C and D on the trajectory. Notably already after approximately 60 % of the total time, the voltage variable v is in its resting state value while it takes the second 40 % of the total time until the second variable h is in its resting state value.*

Table 7.1 – *Distances between points in Figure 7.1 and Figure 7.2.*

	Euclidean distance in v				State time distance [t.u.]				
	A	B	C	D	A	B	C	D	
A	0	1	0.8	0	0	20	190	310	A
B	1	0	0.2	1	20	0	170	290	B
C	0.8	0.2	0	0.8	190	170	0	120	C
D	0	1	0.8	0	310	290	120	0	D

7.1.1 A time based distance measure

This section introduces a measure to quantify the distance between two states s_1 and s_2 for a system without spatial extension. It is based on the time it takes until their trajectories are ε close, i.e. until

$$||s_1(t_i) - s_2(t_j)||_2 < \varepsilon,$$

where t_i and t_j parametrise the respective trajectories in phase space and $||\cdots||_2$ represents the L_2 norm. The distance between two states s_1 and s_2 is then the sum of the times the initial states need to develop to these points:

$$d_t(s_1, s_2) := t_{s_1 \to P_{\mathrm{t1}}^{\mathrm{t2}}} + t_{s_2 \to P_{\mathrm{t2}}^{\mathrm{t1}}}, \tag{7.3}$$

with $P_{\mathrm{t}i}^{\mathrm{t}j}$, $(i, j) \in \{(1, 2), (2, 1)\}$ defined as

$$P_{\mathrm{t}i}^{\mathrm{t}j} := \begin{array}{l}\text{Point in phase space where trajectory } \mathrm{t}_i \text{ of initial state } i \text{ is the first time}\\ \varepsilon\text{- close to the trajectory } \mathrm{t}_j \text{ of initial state } j.\end{array}$$

This idea is explained graphically for the two variable Mitchell Schaeffer model in Figure 7.3 but can be expanded to arbitrary models that develop in time.

For spatially extended systems, the distance is simply the sum of all individual distances:

$$d_t(S_1, S_2) = \sum_i d_t(s_{1,i}, s_{2,i}), \tag{7.4}$$

where $d_t(s_{1,i}, s_{2,i})$ is the distance between the states at the pixel with index i and $S_{1,2}$ denote all states of the respective spatially extended systems.

7.1.2 Application to 2D Mitchell Schaeffer dynamics

I calculated the differences in a 2D spatially extended Mitchell Schaeffer system simulated using the parameters in Table 5.3. It will be compared to a distance based on the voltages:

$$d_{\mathrm{volt}}(v_i, v_j) = \sqrt{(v_i - v_j)^2} \tag{7.5}$$

and to a distance based on the states:

$$d_{\mathrm{state}}(s_{1,i}, s_{2,i}) = \sqrt{||s_{1,i} - s_{2,i}||_2}, \tag{7.6}$$

where $||\cdots||_2$ is the L_2 norm and $s_{1,i}, s_{2,i}$ are the states of the respective system at pixel i. To calculate the timed distance, the phase space of the Mitchell Schaeffer model was discretised into 21 points along each dimension, i.e. $v \in \{0, 0.05, 0.1, \ldots, 1.0\}$

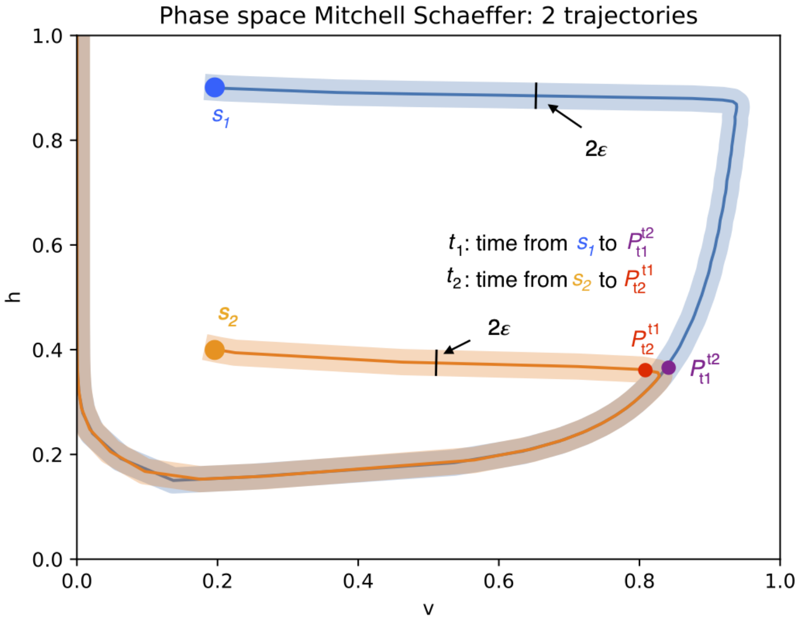
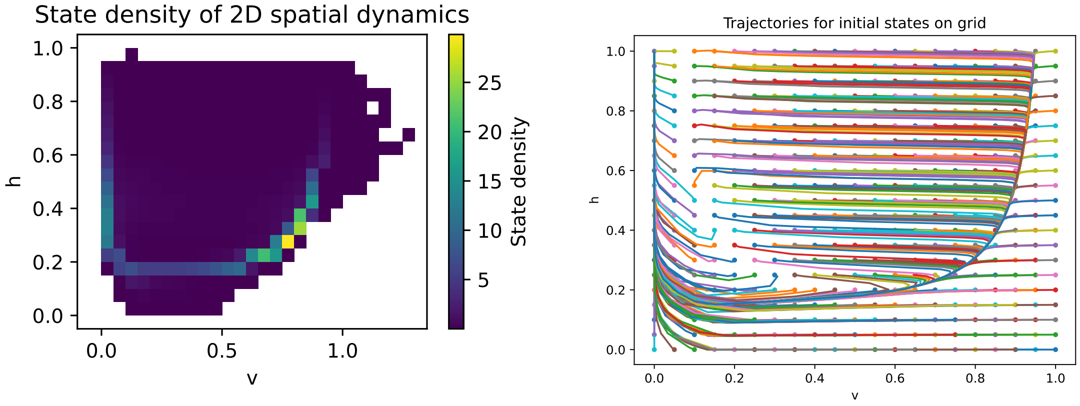


Figure 7.3 – *Explanation of the time based distance measure for two states s_1 and s_2. Their trajectories are developed in time until the respective trajectory comes ε close to the second state's trajectory. For the trajectory that belongs to state s_1 (in blue), this is the case in point $\mathrm{P}_{\mathrm{t1}}^{\mathrm{t2}}$ as can be seen in the graphics. The equivalent point on the trajectory of s_2 (in orange) is $\mathrm{P}_{\mathrm{t2}}^{\mathrm{t1}}$. The timed-distance between s_1 and s_2 is the sum of the times t_1, t_2 it takes from the initial states s_1, s_2 to the points $\mathrm{P}_{\mathrm{t1}}^{\mathrm{t2}}$, $\mathrm{P}_{\mathrm{t2}}^{\mathrm{t1}}$, respectively.*

and $h \in \{0, 0.05, 0.1, \ldots, 1.0\}$ forming a grid of 421 states. Thereafter the distances between each of these states were calculated employing the following algorithm:

1. Take one state combination (s_1, s_2)
2. Calculate the trajectory T_1 of state s_1
3. Check if s_2 is ε-close to trajectory T_1, if yes: $t_2 = 0$, if no:
   - Calculate the trajectory T_2 of state s_2 until it is ε close to T_1. The corresponding time is t_2, the corresponding point is P_{T2}^{T1}
4. Calculate the time at which the distance between trajectories T_1 and P_{T2}^{T1} is minimal using scipy shgo optimiser. This time is t_1.
5. Store this distance as $d(s_1, s_2) = d(s_2, s_1)$.
6. Repeat for all initial state combinations given by the phase space discretisation.

This created a distance table for all states on the 21×21 grid. It has to be calculated once for the corresponding system and the corresponding discretisation. The tolerance has been chosen to $\varepsilon = 0.01$.



(a) **(b)**

Figure 7.4 – *Figure 7.4a: State histogram of the 2-dimensional extended Mitchell Schaeffer dynamics. It was created with a bin width of* 0.05*. The maximum is at state* $s_{\text{mostfreq.}} = (0.75, 0.25)$*. A trace of higher density can be seen at* $h \approx 0.2$ *and* $0.05 \leq v \leq 0.6$ *as well as around the maximum. Figure 7.4b: the trajectories for the discretised state space. The nullcline can be seen: from the top, states between* $v = 0.05$ *and* $v = 0.1$ *and between* $h = 1$ *and* $h = 0.6$ *and then following a curve into the state space do not enter the excitation cycle but decay to the resting state.*

The state of each pixel of the spatially extended system was then approximated by the closest point in the distance table which allowed to calculate the distance between the extended systems as the sum of the local (pixel) state time distances with Equation 7.4. In order to get a better understanding of the values, they were normalised against the time the most frequent state $s_{\text{mostfreq.}}$ in the 2D extended system over time takes to reach the state $s_{\text{timeref}} = (0.05, 0.2)$, i.e. the state when the trajectory enters the refractory phase. To determine this state, the local states s_i of the spatially extended systems were tracked for 9600 tu and plotted as histogram, c.f. Figure 7.4a. The figure shows the most frequent state to be $s_{\text{mostfreq.}} = (0.75, 0.25)$.

The three states used for comparison were chosen to be very close in time with $\Delta t = 1\,\text{t.u.}$ and $\Delta t = 10\,\text{t.u.}$. Figure 7.5a to Figure 7.5c show the states. Figure 7.6a to Figure 7.6b show the distances calculated Equation 7.5, Equation 7.6 and time-based state distance. In all distances the propagation of the excitation wave can be seen as shift of the wave front. For time $t = 421\,\text{t.u.}$ and $t = 425\,\text{t.u.}$ a slightly higher distance can be seen closer to the boundary of the domain. The timed distances show the excitation wave front as region of very high times ($\geq 10\,\text{t.u.}$) and in addition many high-distance pixels all over the tissue. The distance between the states with time $t = 421\,\text{t.u.}$ and $t = 431\,\text{t.u.}$ show a curved line of high distance going outwards from the centre. The time distance does show a high-frequency like pattern in regions of small distance. All distance measures show a higher distance close to the boundary. In the centre all distance measures show a difference between the states.

7.1.3 Discussion and Applications

The results of the 0D case show: there is a blind spot for standard distance measures used in inverse ECG calculations. Some states that are close in time are distant in Euclidean distance measures. The reason for this is that cardiac cell membrane models have to be able to describe the fast upstroke of the cardiac action potential.

First one inaccuracy of the example used in Figure 5.3 has to be discussed. The figure did not show *the next* action potential. Depending on what triggers the system (as there would not be an action potential without an initial stimulus current) the next action potential might start closer in time than the time duration of the potential, i.e. the next start of an upstroke of the action potential (equivalent to point A) could be closer to the end of the action potential (point D). This would result in a smaller time distance between the points A and D in the figure.

Second I have to adress that the results of the 2D case showed a very long timed distance for time frames that where very close in time in the timed distance. The timed distance was at many pixels very high when the actual time difference was only $\Delta t = 1\,\text{t.u.}$. I identified two approaches to explain this: on the one hand states that are very close to the resting state of the local model. As the local model takes a lot of time to relax into the resting state (Figure 7.2), this means even if the states are very close in Euclidean distance in state space, the time until both reach the resting state within the tolerance defined in the time-based distance between them is large. On the other hand states that are on opposite side of the nullcline (the line that separates states that go into the excitation cycle from states that relax into the fixed point without going through the excitation cycle, c.f. Figure 7.4b) have a very large timed distance although they are very close in Euclidean space. This results in a second explanation: two pixels at different sides of the nullcline will have the timed distance of both states coming to the resting states, i.e. a very long time.

This disagrees with the goal I had when I designed the distance measure: states that are in the fixed point and states that just have been excited should have been close together similar to the 1D case. The discrepancy is caused by the initial stimulus current that pushed the resting state over the nullcline in the 0D case but was not taken into account when calculating the distance table in the 2D case. In spatially extended 2D systems the current caused by diffusion pushes the system over the nullcline. This can be seen in the governing equation; it has the form:

$$\frac{\partial v}{\partial t} = \frac{1}{c_m} I_{\text{local}} + D \Delta v,$$

with capacitance c_m the transmembrane voltage v, the diffusion constant D and I the local current density. The equation shows that cells with high transmembrane voltage give a diffusion current to neighbouring cells in their resting state (or at least with a transmembrane voltage variable v smaller than the excited cell's voltage). This behaviour is constitutive for the spatially extended dynamics and has to be represented in the time-based distance measure. It could for example be mimicked for the by

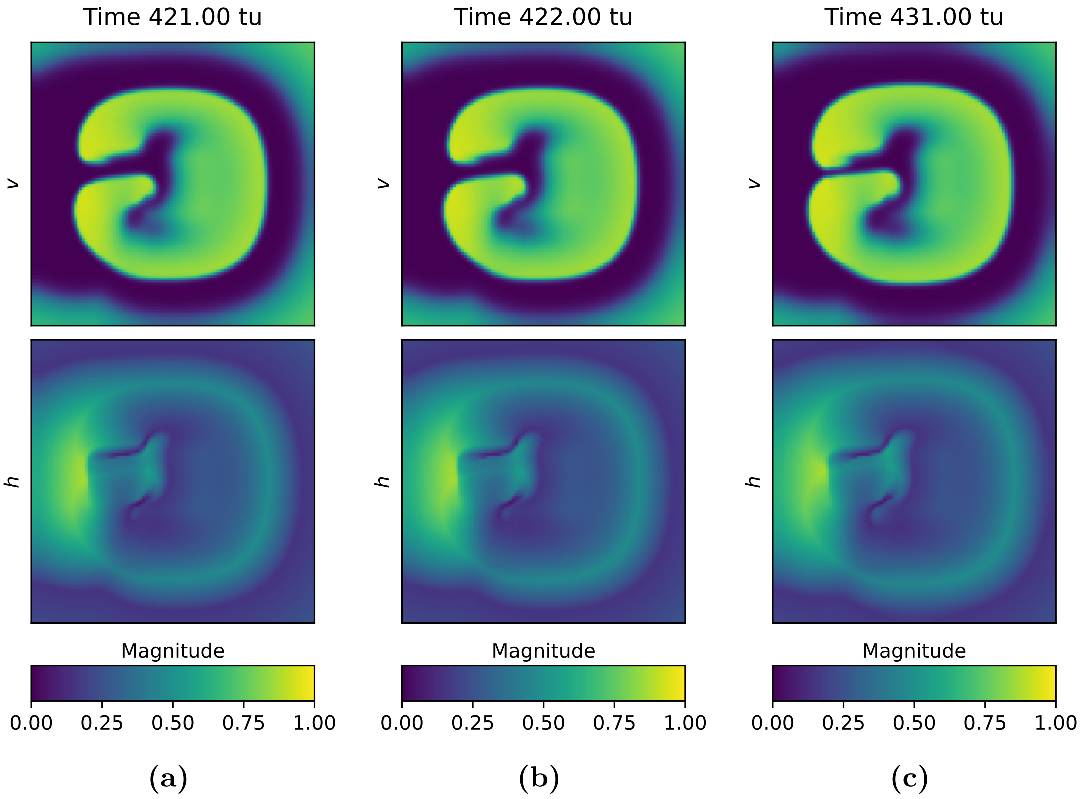
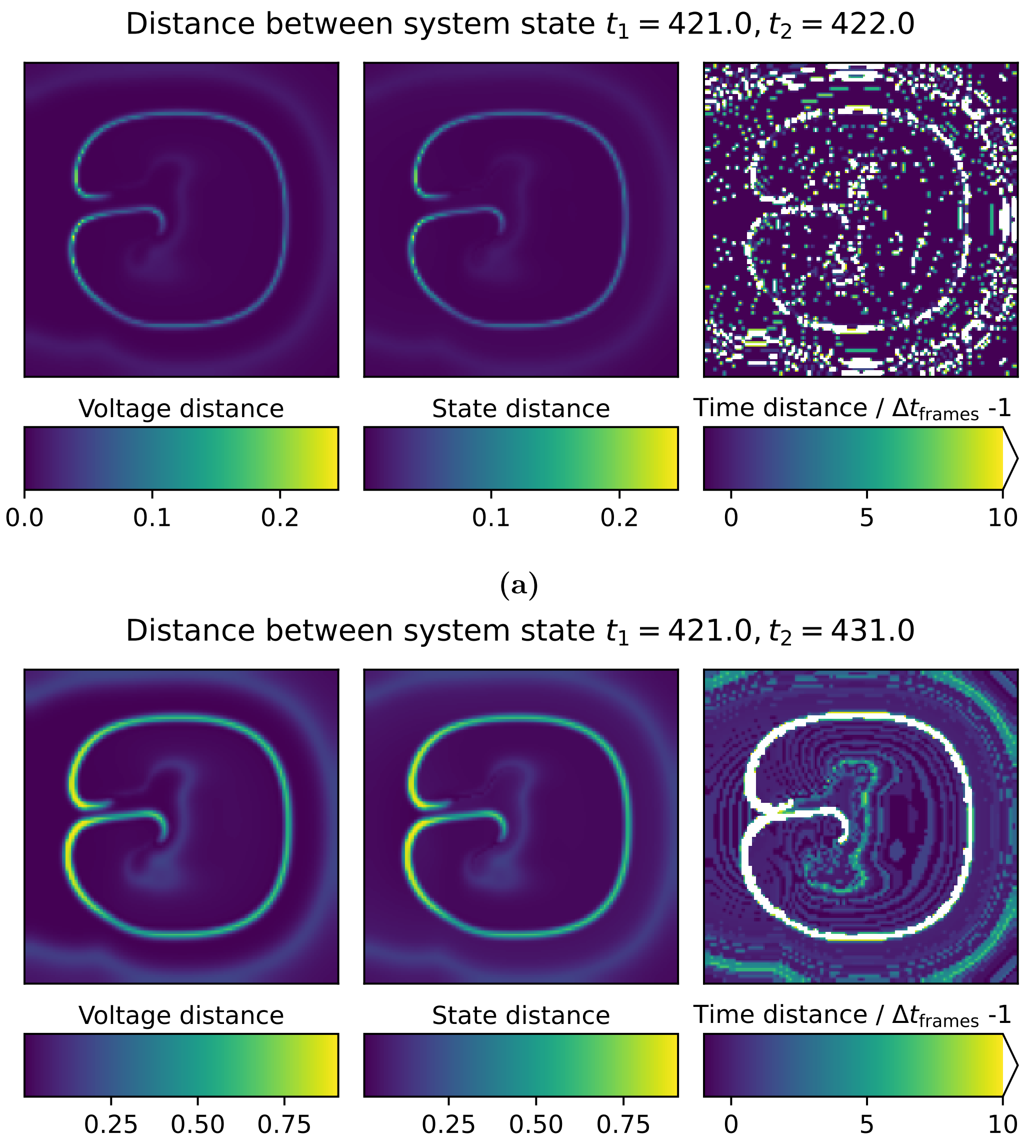


Figure 7.5 – *Three states of the extended Mitchell Schaeffer system used to compare each other with the different distance measures. The upper row depicts the transmembrane voltage at the states while the second row depicts the magnitude of the second variable, h. The time differences are* 1 t.u. *and* 10 t.u.. *This can be related to the dynamics by remembering that the upstroke of the action potential from the resting state is approximately* 20 t.u., *c.f. Figure7.1.*



(a)

(b)

Figure 7.6 – *(continuation) Figure 7.6a — 7.6b: Distances per pixel between the system states at the times indicated. Timed distances larger than 10 × Δt are depicted in white. All distance measures show a distinct line where the excitation wave spreads over the tissue. The timed-distance for $\Delta t = 1$ is very large for many points. This will be discussed in the discussion. All three pattern highlight the same features: one large connected wave front that spreads from the centre similar to what can be expected based on Figure 7.5a — 7.5c.*

calculating the distance table with a small initial stimulus.

A second approach to adress this effect of the spatially extended system being effectivly oscillation-like due to the diffusion could make use of an oscillator equation that oscillates in a form suitable for cardiac local dynamics: one fast excitation cycle and one slow recovery phase. This approach will need to map the pixel's state onto the oscillator phase space.

Another open challenge is that often only the transmembrane voltage variable v is accessible in experiments. This would require state reconstruction based on the transmembrane voltage time series as discussed by Herzog et al. [137].

These two approaches work very well together. It should be possible to reconstruct a state in the oscillator model based on the transmembrane voltage variable — another open research question that emerged from this idea. This allows to make use of this method in real experiments where the underlying dynamics of the heart are never fully known. It would therefore convert the problem from the real world into the model world.

The timed distances for the 2D spatially extended systems showed a wave like pattern in regions of small distances. I suppose that reason for this is the discretisation of the phase space. The discretisation cell size was $\frac{1}{20}$ of the value range. It is not clear whether this is good enough or not.

The initial scans of the phase space will need some time. However they only have to be done once per system, thereafter the nearest neighbour look up will provide the distance without new calculations. The region in state space that has to be scanned could first be narrowed down by making a phase space histogram for the dynamics to investigated as I have done in Figure 7.4a.

Another open question is how to normalise the timed distance to give it a more intuitive meaning. For oscillator-like behaviour the time of one oscillation appears to be convenient. Another option could be the duration of the excitation phase. One has to think about what a meaningful normalisation means in this context.

There are still many open questions. Nevertheless I believe this distance measure is worth further investigation.

Applications

Distance measures are used everywhere in reconstruction: They are used to compare a reconstruction with a true solution. They are used in cost functions of machine learning algorithms to guide the algorithm's learning phase.

The Euclidean measures give a large error if the wave front of an excitation wave is shifted in the reconstruction. Even though, as shown, the pixel states are not *really* different. In the example I showed the same was true for the time based method, however if a real oscillator will be used to calculated the time based distance this will probably disappear and it will be easier to estimate the quality of a reconstruction.

A second area of application is in machine learning. This approach might smooth the gradient space in gradient descent learning as it distinguishes values from each other that previously couldn't be distinguished before (does a small v-value mean resting state or refractory state?) and as it provides to 'navigate' through phase space as it gives a different answer to the question: 'which states are close?'

7.2 Referencing the ECG in time

Modern cardiac medicine is difficult to imagine without accessing the electrical state of the heart by means of the ECG [35, 36, 138]. Huge effort is done to relate patterns in the ECG to underlying cardiac electrical states, to access information about the state of the heart and to decide on treatment including deep learning and other computational methods [139, 140]. The origin of the ECG signal is the cardiac extracellular potential, however in its very nature as a potential accessing it can only be measured against a reference [141]. In medical diagnosis, the reference (*lead electrode*) is either another electrode placed at the body of the patient or a combination of electrodes forming a so called *virtual electrode*[1] [142]. The most common one is the *Wilson's central terminal* (WCT) which averages the signal of three electrodes placed on the patient and uses this average as a reference.

However already in 1955 Burger [143] showed, that the WCT cannot be seen as a zero reference as referencing to infinity would be. Frank [144] showed in 1954 that the WCT-reference is expected to vary over time.

Since then efforts are made to understand the error introduced by using the WCT in ECG-measurements [145–148] and efforts were taken to derive a zero potential based on the measurements [149]. In 2018 Moeinzadeh et al. [150] tried to minimise the fluctuations of the WCT in time by finding (constant) weights for the individual measurement electrodes with a genetic algorithm. While they could reduce the WCT amplitude over time by one order of magnitude, the corresponding weight factors varied among the individual patients, i.e. it was not possible to find a generic solution for the problem of ambiguous ECG signals due to the variation in the signal either being originated at the (virtual) reference or at the electrode itself.

Gargiulo presented 2015 a measurement device that references an ECG electrode signal against a (in time) low-pass filtered signal of itself [151]. I have studied this approach numerically. In Section 7.2.1 it will be shown for simplified 2D tissue models that the ECG signal calculated with a WCT equivalent as reference fails to follow the underlying cardiac dynamics due the averaging.

7.2.1 Application of referencing methods in 2D simulations

Justine Wolter provided simulation data of a monodomain simulation for the voltage variable v_m with the Fenton Karma model [152] on a 180×180 pixel discretised in time in 140 time steps. Three electrode signals ϕ_i^n at time step n were calculated as average of the v_m values on three 4×4 pixel to model the ECG data.

[1] The term is ambiguous in term of defibrillation research, where virtual electrodes also denotes sources of electrical activity on and in heart tissue caused by defibrillation currents.

Table 7.2 – *Parameters to reference in time in 2D simulations.*

Num electrode	x_0	y_0	w_x	w_y
1	60	40	4	4
2	60	140	4	4
3	140	90	4	4
Butterworth filter	N		cutoff	
	2		0.1 f_{Nyquist}	
Sliding window	w_t			
	25 frames			

The WCT was calculated by taking the mean of the three electrode signals at the corresponding time step

$$\phi^n_{WCT} = \frac{1}{3}(\phi^n_1 + \phi^n_2 + \phi^n_3). \tag{7.7}$$

Two alternative reference concepts were implemented. In the first one, the reference is calculated against the mean value of a sliding window in time of size w_t over the past values of that single electrode, i.e. each electrode defines a reference signal for itself by its past values. For electrode i at time step n the reference signal $\phi^n_{\text{sl},i}(w_t)$ with sliding window width w_t in units of time steps is calculated by

$$\phi^n_{\text{sl},i}(w_t, \vec{\phi_i}) = \frac{1}{w_t} \sum_{\tau=n-w_t+1}^{n} \vec{\phi_i}^{\,\tau}, \tag{7.8}$$

where $\vec{\phi_i}^{\,\tau}$ is the τth component of the vector $\vec{\phi_i}$ that represents the time series of electrode values of electrode i discretised in time. The second concept also references in time but using a butterworth filter B^N_{cutoff} [153] with N representing the filter order and cutoff the filter's cut off frequency in the scipy [110] implementation. The reference signal for electrode i at time step n becomes

$$\phi^n_{B,i}(\vec{\phi_i}) = B^N_{\text{cutoff}}(\vec{\phi_i}^{\,n}), \tag{7.9}$$

where B^N_{cutoff} describes the butterworth filter of order N and with cut off frequency cutoff at time step n and $\vec{\phi_i}^{\,n}$ is the discretised time series of the electrode values of electrode i up to the current time step n. Note that the filter is only provided the electrode values up to the current time n. Future values are not used.

All parameters used are listed in Table 7.2. Figure 7.7b compares the three reference signals for electrode 3 (depicted magenta in Figure 7.7a which also shows a frame of the dynamics). It can be seen that the butterworth filter does not return values from the beginning. The signal follows the dynamics of the electrode signal but also shows a trend slowly moving to the electrode signal. The sliding window referenced ECG also shows a trend before following only the signal. This is caused by the implementation which if the sliding window exceeds before the first frame repeats the value corresponding to the first frame. This also explains why the value for frame 0 is 0. In contrast to these two time-based reference methods the WCT reference shows fluctuation while following the ECG signal. Reason for this is the influence of the other electrodes' measurements (see below). Based on this comparison for the rest of this section the sliding window reference has been used to reference in time as it shows a smaller and temporally limited trend.

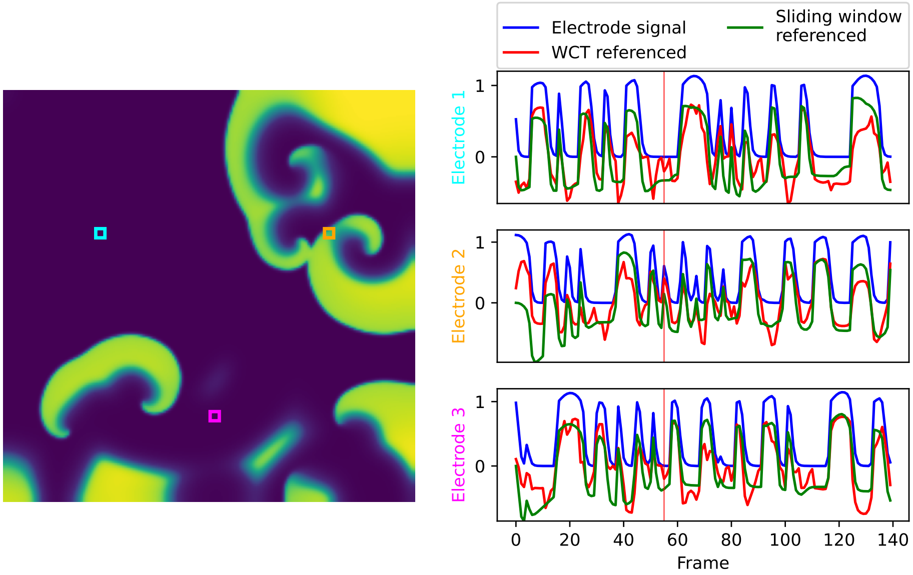
To better understand the WCT behaviour, one spatially extended frame of the dynamics with the electrodes indicated in colour in space and the electrode signals is given in Figure 7.7a. The x and y values given in Table 7.2 are row-values (x) and column values (y) resulting in the cyan electrode being electrode 1, the orange electrode being electrode 2 and the magenta electrode being electrode 3. The left part of the figure shows the frame for time step 55. The dynamic shows spiral wave behaviour. The ECG electrode value is 0 in electrodes 1 and 3 and approximately 0.6 for electrode 2. This is also shown on the right side of the figure. It shows the time series of the electrode signal, the WCT-referenced signal and the sliding window referenced signal. A red vertical line marks time step 55. At that point in time $\phi_1^{n=55} = 0$ and $\phi_3^{n=55} = 0$ but $\phi_2^{n=55} \approx 0.6$, leading to not only the WCT referenced signal $\phi_2 - \phi_{WCT}^{n=55}$ being very close to the electrode value but also causes a change in direction from increasing to decreasing in electrode 1. In comparison the sliding window referenced value does not show this behaviour as the value does not depend on the other electrode signals. Note that this may affect measures sensible to fluctuations of the ECG like for example the permutation entropy employed in [154].

7.2.2 Application in Langendorff perfusion simulations

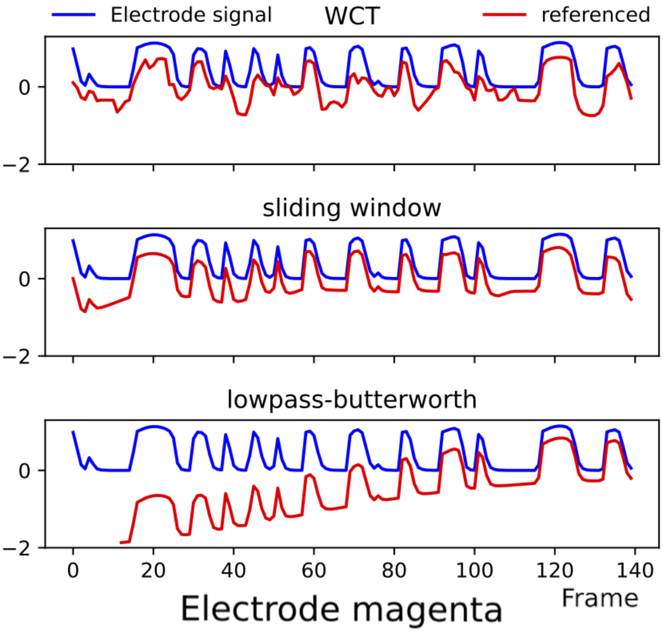
Section 7.2.1 showed that spatial referencing introduces fluctuations to the ECG signal that are not related to tissue close to the electrode. Can this effect also be seen in more realistic models? I examined this by (A) refining the spatial model to the experiment's geometric model (B) calculating a 'real' ECG signal. This means I reconstructed the potential following the approach presented in Section 5.2.3.1, I modelled the diffusion into the Langendorff bath and I calculated the electrode signals.

The cardiac dynamics was simulated with the Mitchell-Schaeffer model as local model. The parameters of the model were $\tau_{\text{close}} = 150$, $\tau_{\text{open}} = 120$, $\tau_{\text{out}} = 6$, $\tau_{\text{in}} = 0.3$, $v_{\text{gate}} = 0.13$. The conductivity of the heart tissue was chosen to $\sigma_i = 1 \times 10^{-3}$. The time domain was discretised with $\Delta t = 0.1$. The parameter $\lambda = 0.75$ was chosen. Remember, λ is the ratio between the intracellular conductivity and the extracellular conductivity, σ_i and σ_e respectively, in the monodomain approximation (c.f. Section 2.5). The conductivity in the Langendorff bath was set to $\sigma_b = 1$.

Figure 7.8 shows the geometrical setup and labels that will be used in the following description. Each of the four electrode panels in the figure has 35 electrodes, i.e. there were 140 electrode signals in total. The simulation ran until $t_{\text{end}} = 5000\,\text{t.u.}$. For each electrode I calculated the WCT-reference signal based on the four electrode indicated in black in the Figure 7.8. The solver of the potential reconstruction reconstructs the



(a)



(b)

Figure 7.7 – *Figure 7.7a: electrode signal, sliding window referenced signal and WCT-referenced signal from the electrodes marked in colour in the snapshot from the dynamics on the left. Figure 7.7b: comparison of the reference methods. The sliding window method (middle) and the low-pass butterworth filter exhibit a trend after which they almost exactly follow the electrode signal. The butterworth filter only starts performing after a certain time, therefore it starts giving a signal at frame 15. Note that the WCT reference signal does not follow the electrode signal but exhibits to the electrode signal unrelated fast changes, e.g. in frames 100 to 120. This is caused by the signals of the other electrodes used to calculate the reference, see Figure 7.7a.*

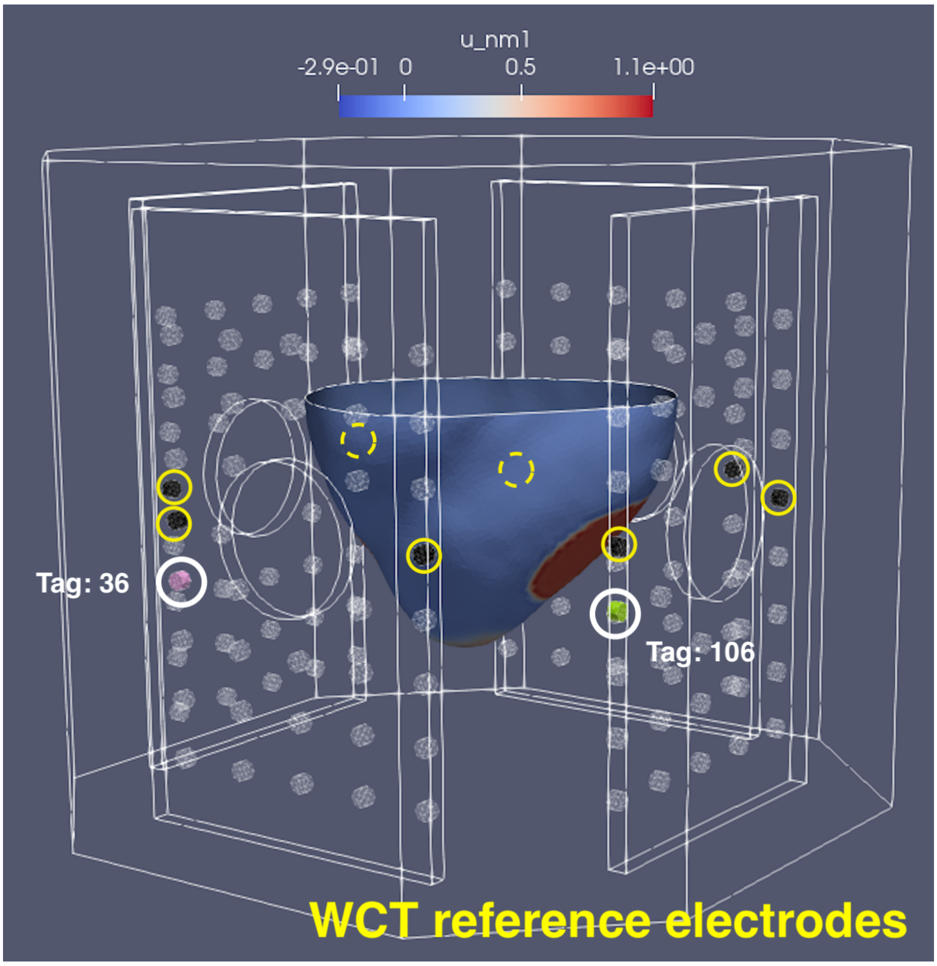
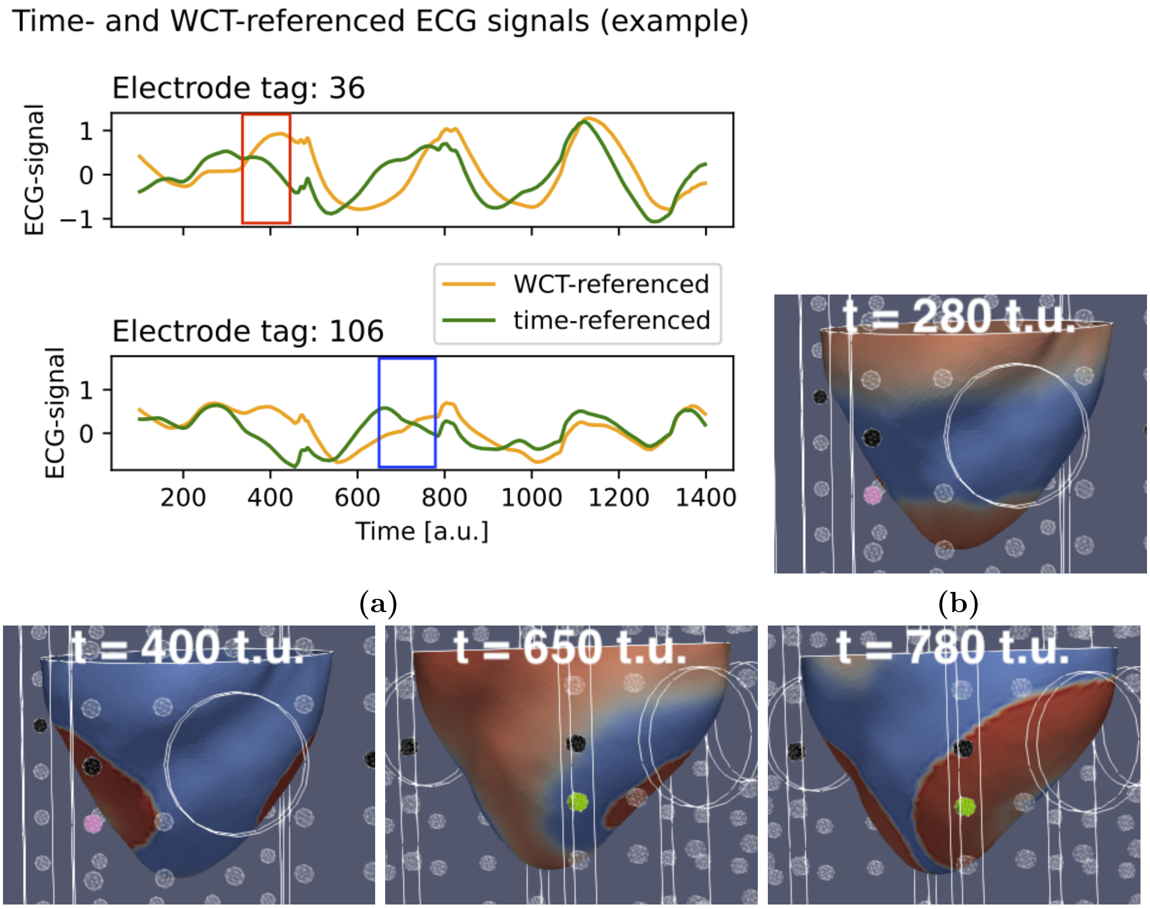


Figure 7.8 – *Geometrical model of the simulation domain. The electrical dynamics has been simulated on the heart surface. The same WCT electrodes have been used that were used in the experiments. This are 8 electrode drawn in black and indicated in yellow. All are on the same height. Two reference electrode cannot be seen as they are behind the heart. They are indicated in yellow with dotted line style. Two electrodes are marked in white. Their signal will be used exemplarily. On the lower right of the heart an excitation wave can be seen.*



(a) (b)

(c) (d) (e)

Figure 7.9 – *Figure 7.9a: Two regions of different behaviour of the ECG signals are indicated. The upper panel shows the electrode signal of the rose electrode in Figure 7.8. In the time interval indicated by the red rectangle the WCT-referenced electrode signal increases while the time-referenced signal decreases. Significant different behaviour is also found in the lower panel (green electrode) for the time interval indicated by the blue rectangle: The WCT-referenced signal is increasing in that region while the time-referenced signal decreases. Figure 7.9b: The end of a wave that dissolve into the upper boundary. Figure 7.9b: The area close to the electrode is not yet excited therefore the time-referenced ECG signal still decreases. Figure 7.9d — 7.9e: While the wave (left in Figure 7.9d) is travelling away from the electrode the timed reference decreases until (Figure 7.9e) the new excitation wave comes close to the electrode.*

potential up to a constant, therefore the ECG signals used for the time-referenced frame were referenced in each time step against the mean over all electrodes. Thereafter I calculated for each electrode the sliding window reference with a window width of $w_t = 70\,\text{frames} = 140\,\text{t.u.}$.

Figure 7.9 shows exemplary ECG time series with substantial differences. Indicated in colour are two regions where the time-reference and the WCT-reference differ significantly; while one of the time series decreases, the other increases. Some pattern are present in both reference frames. At some positions (e.g. time 900 t.u. and time 1000 t.u.) one ECG-signal shows the same form as the other. Figure 7.9b — 7.9e show three exemplary heart dynamic states from the perspective of the electrodes. The times were chosen such that they are approximately in the time window of the times of the time windows marked in Figure 7.9. They show at time $t = 280$ t.u. a wave that disappears when it hits the no flux boundary on the upper part of boundary while there is a new wave arising from the bottom. At time $t = 400$ t.u. in the centre of the extract there is no activity, at the sides of the cutting there is excited tissue. At time $t = 650$ t.u. a large part of the heart tissue at that region is excited while on the right there is a non-excitation region coming. In the cutaway for time $t = 780$ t.u. the tissue very close to the electrode is being excited.

In this examples the time referenced signal does not follow the excitation of the heart but the WCT-referenced signal does.

7.2.3 Discussion

In the generic 2D simulation the ECG signal of a WCT-referenced electrode could not be interpreted as if it gave information about the dynamics at the electrode. This effect is not surprising as the referenced signal maps electrode activity from different regions on the heart. However this is hardly reflected in today's research as I have pointed out in the introductory part of this section.

I have also compared two time based measures, on low-pass filter and a sliding window reference. Based on the results shown in Figure 7.7b a sliding window approach seems to be more suitable. It is not instantly clear how to choose the length of the sliding window. Considering that one wants to have a stable reference, a large window is suitable.

The 3D simulations showed a difference between the two ECG reference method but the time series shown in Figure 7.9 together with the excitation waves show that the WCT-referenced signal follows the dynamics of the heart while the time-referenced signal does decrease when a wavefront is moving towards the electrode. I expect this to be an artefact from having chosen a time window that is too small. The timewindow $w_t = 140$ t.u. does not cover one period in the ECG signal, time averaging therefore cannot average changes in the ECG out.

Looking at the ECG traces shown it appears that sometimes there is a shift in time between the signal. For example between times 800 t.u. and 1000 t.u. for reference electrode 36.

I used the Generalized Minimal Residual method in the PetSc implementation which handles that the potential is only defined up to a constant automatically[2]. This is done in each time step. As a consequence there is no reason to expect this constant chosen by the solver to be constant over time. As a result for this solver setup a reference in space had to be done in order to reference the electrode values against each other.

A fixed constant could be achieved by defining one set of cells to have a fixed value and imposing this as Dirichlet boundary condition. Then every potential would be automatically measured against this. This has been discussed for the bidomain equation [155]. The solver was much slower in that case. However it might be necessary to substantiate the findings presented in this section.

Another point for future work is a mesh density analysis as within the time of this work I couldn't check how the results change with finer meshes. However I do not expect substantial changes as one might loose features due to too coarse mesh resolution but in general much of the form of the solution of the mathematical problem is reproduced by the finite element method, c.f. Figure 2.3.

This concept that I have developed could bring additional information for ECGI methods to improve the data basis used to perform the reconstruction.

[2] The user 'tells' the solver for which entries this is expected and the solver handles this on its own.

Chapter 8

Conclusions and outlook

8.1 Medical image processing and numerical framework

I implemented a framework with which based on medical image data I can create geometrical models to perform simulation on them. The image segmentation process was done by hand, however future work could — similar to [156] — make use of the fractional anisotropy (FA) to guide segmentation. FA is a measure to characterise how anisotropic a voxel is based on its diffusion tensor. I have only very recently implemented a semi-automatic way to find the mesh size parameters for which the FEM solution does not depend on the spatial discretisation. This should be checked for the simulations for which it has not yet been done. However even the not-checked results can be used to provide conclusions as the FEM solution at the discretisation cell boundaries is very precise (see Figure 2.3).

I had to find and learn various tools to implement this framework, I used 3D Slicer to do segmentations (distinguish different types of tissue) in medical image data. I found and learned MeshLab to do post analysis of the resulting surface meshes. I used and learned Gmsh to model the experimental setup, the Multi-ECG panel and generic shock electrodes. Within Gmsh I implemented rotation, translation and scaling of the geometric models of the heart and furthermore automatic centralising of the heart between the shock electrodes in current density experiments. This enabled Simon Wassing and me to perform the current density experiments and me to perform first electric activity simulations.

The registration work that was done by Benjamin Weiß and mentioned in Section 4.1.2 could later be made use of to combine C-arm data that provide a good resolution of the bone structure and MRI-data that resolve the heart and other tissues like the lungs and, depending on the measurement type, the fibre directions in the heart. Since I have programmed all the necessary components in a fashion that based on a segmentation and information on the translation needed the process to obtain a volumetric boundary mesh is quasi-automatic (only parameters like mesh cell size have to be chosen during the process) including interpolation of the fibre orientation data onto the numerical grid, future students can (with little instructions) easily use this workflow framework.

Special attention should be given to the feature of the continuous current formulation described in Section 5.2.2. To my knowledge, this thesis first makes of use of this approach in cardiac defibrillation current simulations. It allowed us to use the current as main observable in Chapter 6.

8.2 Current density analysis

Simon Wassing and I have simulated the Langendorff perfusion setup that is used in the experiments in the group and found: in Langendorff perfusion setups similar to the one we use in the research group biomedical physics depending on the heart size between 85 % to 92 % of the total current applied in defibrillation experiments avoided the heart tissue for a heart centralised between the electrodes. For two hearts with different size this means: if the same voltage difference is applied over the shock electrodes, the current that penetrates the heart is different, i.e. the effective current at hand for defibrillation varies with the heart size if the same voltage is applied.

This has an effect on the comparability of experiments. Simon Wassing has measured the weight of hearts before and after the experiment and quantified that their weight increases, reassuring a hypothesis that the heart soaks itself with water over the experiment. As the weight increased also the volume of the heart increases and as a result the defibrillation current applied to the heart at different times in an experiment varies even if the voltage is held constant. It would be important to measure how large the difference in volume of the hearts is to see (then in simulations) how much the effective current for defibrillation varies within one experiment.

We also found out that the position of the heart with respect to the electrodes affects how much current is delivered to the heart. Figure 6.2 shows that moving the heart downwards from the centralised position increases the ratio of current accessible for defibrillation. As the heart is conical shaped by moving it downwards from the centralised position between the electrodes effectively the cross section of the heart with the current field increases. This leads to a higher current through the heart.

We found that the current through the heart I_h normalised by the volume to the power 2/3, $V^{2/3}$ is roughly constant. In his Master's thesis [98] Simon Wassing developed an equation which allows to find an equivalent voltage difference over the electrodes to supply two hearts with different sizes with the same defibrillation current. For heart 1 with volume $V(\alpha_1)$, where α is a size factor, and heart current $I_{\mathrm{h},1}$ induced by voltage difference U_1, in order to obtain the same current through the heart two with Volume $V(\alpha_2)$ and $I_{\mathrm{h},2}$, the voltage difference U_2,

$$U_2 = U_1 \left(\frac{V(\alpha_1)}{V(\alpha_2)} \right)^{2/3}.$$

We compared defibrillation currents in isotropic hearts (without fibre orientation) with defibrillation currents of anisotropic hearts (with fibre orientation). We found that for the same current through the heart, the current density distribution of simulations with fibre orientation is broader and covers also a larger range into higher current densities. This means that fibre orientation can possibly *increase* the likelihood of a successful defibrillation in simulations because higher current densities are present in the heart. In other words, simulations without fibre orientation possibly underestimate defibrillation probability.

I have finished the framework in a way that also in-vivo simulations can be conducted within this framework and simulated the current density distribution for three different ICD positions. The next step in this in-vivo setup will be to adjust the conductivities to conductivities from literature and to use the fibre-orientation data in this simulation.

8.3 New concepts

This is almost the end of this thesis and you haven't seen a single electrical dynamic reconstructed from an ECG signal — Chapter 7 is the reason. When I tried to understand the problem with the inverse ECG I found two parts of the puzzle that I consider missing: 1. Euclidean-like measures used to compare two states have a systematic error which is not accounted for → Is there a need for a new distance measure? 2. The voltage measured at an ECG electrode in the standard setup is ambiguous as it is referenced against other electrode's signals. As a consequence reconstruction based on the ECG signal is only defined up to a constant. All tissue excited and all tissue in resting state would have the same ECG signal. → Is there an alternative reference?.

In the first section of that chapter, Section 7.1 I addressed the first of the two questions and I provided a possible solution. I made use of the fact that the heart is an excitable medium, i.e. it is made of locally excitable cells that are coupled through diffusion and defined the distance between two local cells based on the time their local states need to come very close to each other in state space. I have shown in that respective section how this conceptually works and applied this distance measure to 2D extended Mitchell Schaeffer dynamics. A more detailed discussion is given in Section 7.1.3.

In the second section of this chapter, Section 7.2 I present an idea how the ECG electrodes could be referenced against a mean value in *time* (and not in space). The intuition behind this is that if the time windows used for referencing is large enough this value is approxitmately constant and variations in the ECG-electrode signal would (almost) uniquely be attributable to that electrode.

I have shown that a time based reference follows the transmembrane voltage dynamics at one electrode signal better than the spatially referenced electrode signal for a simple model of ECG measurements in a 2D spatially extended system in Section 7.2.1. Then I employed the surface dynamics simulation in the Langendorff bath to research if this effect can be seen in a more realistic setup. In the Langendorff bath simulations however the WCT-referenced signal followed better the underlying transmembrane voltage dynamics. I drew as conclusion that this was because the reference-time window was too small. This was discussed in Section 7.2.3.

After there has been no research on the ECG configuration since approximately the 1960s, beginning the years 2012 a group in Australia already followed that idea and discussed several approaches to implement a time-based ECG-reference via a low-pass filter [151, 157].

8.4 Good scientific digital practices

In Chapter 3 I have presented a rather personal project: I presented how I have learned to think about my daily scientific work. Now, as it has emerged I wonder how I organised my work. My key was to think of my daily tasks as *scientific activities* and whenever I start something I first name it: Am I doing data analysis? Am I visualising something? Am I trying things out (as it is also necessary in scientific work)?

Then I sorted the activity to a project on which I am working. If there is no project for which this is done, is it really interesting? And if I think yes but there is no project, I created a project. Because goals are organised in projects and it helps me to think of what I was doing something for.

Surely every person is different and some might not actually need to think of scientific work that way, however I am a personal that needs to. The boost in productivity and confidence in what I was doing and in focussing on my goals was enormous. This casually also helped me foster organising my work traceable and reproducible as I made use of this structure also on the file system.

I believe concepts of how to not loose the bigger picture in the everyday work should be taught at university as scientists very often do both, thinking of the bigger picture *and* to the daily work.

8.5 Outlook

The current density simulations have shown that the positioning of the heart in low-energy defibrillation with respect to the electrodes matter. I think there are two directions which one could follow. In the end it will be important to understand how the current controls the arrhythmia, therefore one could simulate one electrical pattern with currents for varying electrode positions. The other direction would be to test this in experiments using current density imaging techniques.

I think that I have found two new concepts which I believe have the potential to add important extra information for state reconstruction methods for the ECGI. The next step is to test this hypothesis, i.e. take the timed state distance into account in the cost function of a state reconstruction method and make use of time-referenced ECG-signals as input for ECGI methods.

I have developed a work flow on how to go from medical data to simulation results in this thesis, however after all that work I believe it might be more future proof to join shared efforts like for example the CARP project developed in Graz. The computational science work to maintain a simulation framework up to date, efficient (as it is needed for whole heart simulations) and tested is enormous. However the knowledge I gained documented in this thesis will help future users.

The timed-ECG reference was used by the group in Australia trying to improve today's clinical ECGs. I don't think that this will work because millions of doctors have learned how to work with the old ECG and there is a similar amount of literature how to interpret ECG-signals. However I am not aware of anyone who has used this with the inverse ECG exploiting the idea that by referencing in time one source of uncertainty is removed.

References

[1] *International Society for Heart Research*. https://ishrworld.org/.

[2] S. A. Niederer, J. Lumens, and N. A. Trayanova. “Computational Models in Cardiology”. In: *Nature Reviews Cardiology* 16.2 (Feb. 2019), pp. 100–111. DOI: 10.1038/s41569-018-0104-y.

[3] S. M. Sidik. “Heart-Disease Risk Soars after COVID — Even with a Mild Case”. In: *Nature* 602.7898 (Feb. 2022), pp. 560–560. DOI: 10.1038/d41586-022-00403-0.

[4] B. M. Ogle et al. “Distilling Complexity to Advance Cardiac Tissue Engineering”. In: *Science Translational Medicine* 8.342 (June 2016), 342ps13–342ps13. DOI: 10.1126/scitranslmed.aad2304.

[5] *Cardiovascular Diseases*. https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1.

[6] *Cardiovascular Diseases (CVDs)*. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[7] J. Berdowski, M. ten Haaf, J. G. Tijssen, F. W. Chapman, and R. W. Koster. “Time in Recurrent Ventricular Fibrillation and Survival After Out-of-Hospital Cardiac Arrest”. In: *Circulation* 122.11 (Sept. 2010), pp. 1101–1108. DOI: 10.1161/CIRCULATIONAHA.110.958173.

[8] R. A. DeSilva, T. B. Graboys, P. J. Podrid, and B. Lown. “Cardioversion and Defibrillation”. In: *American Heart Journal* 100.6, Part 1 (Dec. 1980), pp. 881–895. DOI: 10.1016/0002-8703(80)90071-X.

[9] C. Hanefeld. “A First City-Wide Early Defibrillation Project in a German City: 5-Year Results of the Bochum against Sudden Cardiac Arrest Study”. In: *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 18 (June 2010), p. 31. DOI: 10.1186/1757-7241-18-31.

[10] M. C. Baumgarten, J. Röper, K. Hahnenkamp, and K.-C. Thies. “Drones Delivering Automated External Defibrillators—Integrating Unmanned Aerial Systems into the Chain of Survival: A Simulation Study in Rural Germany”. In: *Resuscitation* 172 (Mar. 2022), pp. 139–145. DOI: 10.1016/j.resuscitation.2021.12.025.

[11] *Wo finde ich einen Defibrillator?* mydefibri.com. July 2014.

[12] A. Brandes, M. D. Smit, B. O. Nguyen, M. Rienstra, and I. C. Van Gelder. “Risk Factor Management in Atrial Fibrillation”. In: *Arrhythmia & Electrophysiology Review* 7.2 (June 2018), pp. 118–127. DOI: 10.15420/aer.2018.18.2.

[13] D. H. Lau, S. Nattel, J. M. Kalman, and P. Sanders. “Modifiable Risk Factors and Atrial Fibrillation”. In: *Circulation* 136.6 (Aug. 2017), pp. 583–596. DOI: 10.1161/CIRCULATIONAHA.116.023163.

[14] B. J. J. M. Brundel, X. Ai, M. T. Hills, M. F. Kuipers, G. Y. H. Lip, and N. M. S. de Groot. “Atrial Fibrillation”. In: *Nature Reviews Disease Primers* 8.1 (Apr. 2022), pp. 1–23. DOI: 10.1038/s41572-022-00347-9.

[15] U. Schotten, S. Verheule, P. Kirchhof, and A. Goette. “Pathophysiological Mechanisms of Atrial Fibrillation: A Translational Appraisal”. In: *Physiological Reviews* 91.1 (Jan. 2011), pp. 265–325. DOI: 10.1152/physrev.00031.2009.

[16] D. D. Berg, C. T. Ruff, and D. A. Morrow. “Biomarkers for Risk Assessment in Atrial Fibrillation”. In: *Clinical Chemistry* 67.1 (Jan. 2021), pp. 87–95. DOI: 10.1093/clinchem/hvaa298.

[17] M. Glikson and P. A. Friedman. “The Implantable Cardioverter Defibrillator”. In: *The Lancet* 357.9262 (Apr. 2001), pp. 1107–1117. DOI: 10.1016/S0140-6736(00)04263-X.

[18] M. Mirowski. “The Automatic Implantable Cardioverter-Defibrillator: An Overview”. In: *Journal of the American College of Cardiology* 6.2 (Aug. 1985), pp. 461–466. DOI: 10.1016/S0735-1097(85)80186-8.

[19] F. Godemann, C. Butter, F. Lampe, M. Linden, M. Schlegl, H.-P. Schultheiss, and S. Behrens. “Panic Disorders and Agoraphobia: Side Effects of Treatment with an Implantable Cardioverter/Defibrillator”. In: *Clinical Cardiology* 27.6 (2004), pp. 321–326. DOI: 10.1002/clc.4960270604.

[20] A. C. Thomas, S. A. Moser, M. L. Smutka, and P. A. Wilson. “Implantable Defibrillation: Eight Years Clinical Experience”. In: *Pacing and Clinical Electrophysiology* 11.11 (1988), pp. 2053–2058. DOI: 10.1111/j.1540-8159.1988.tb06349.x.

[21] J. E. Poole et al. “Prognostic Importance of Defibrillator Shocks in Patients with Heart Failure”. In: *The New England journal of medicine* 359.10 (Sept. 2008), pp. 1009–1017. DOI: 10.1056/NEJMoa071098.

[22] D. Hornung, V. N. Biktashev, N. F. Otani, T. K. Shajahan, T. Baig, S. Berg, S. Han, V. I. Krinsky, and S. Luther. “Mechanisms of Vortices Termination in the Cardiac Muscle”. In: *Royal Society Open Science* 4.3 (), p. 170024. DOI: 10.1098/rsos.170024.

[23] V. Kappadan, S. Telele, I. Uzelac, F. Fenton, U. Parlitz, S. Luther, and J. Christoph. “High-Resolution Optical Measurement of Cardiac Restitution, Contraction, and Fibrillation Dynamics in Beating vs. Blebbistatin-Uncoupled Isolated Rabbit Hearts”. In: *Frontiers in Physiology* 11 (May 2020). DOI: 10.3389/fphys.2020.00464.

[24] J. Christoph et al. “Electromechanical Vortex Filaments during Cardiac Fibrillation”. In: *Nature* 555.7698 (Mar. 2018), pp. 667–672. DOI: 10.1038/nature26001.

[25] P. Bittihn, A. Squires, G. Luther, E. Bodenschatz, V. Krinsky, U. Parlitz, and S. Luther. “Phase-Resolved Analysis of the Susceptibility of Pinned Spiral Waves to Far-Field Pacing in a Two-Dimensional Model of Excitable Media”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1918 (May 2010), pp. 2221–2236. DOI: 10.1098/rsta.2010.0038.

[26] S. Luther et al. “Low-Energy Control of Electrical Turbulence in the Heart”. In: *Nature* 475.7355 (July 2011), pp. 235–239. DOI: 10.1038/nature10216.

[27] F. H. Fenton, S. Luther, E. M. Cherry, N. F. Otani, V. Krinsky, A. Pumir, E. Bodenschatz, and R. F. Gilmour. “Termination of Atrial Fibrillation Using Pulsed Low-Energy Far-Field Stimulation”. In: *Circulation* 120.6 (Aug. 2009), pp. 467–476. DOI: 10.1161/CIRCULATIONAHA.108.825091.

[28] *Ablation for Arrhythmias.* https://www.heart.org/en/health-topics/arrhythmia/prevention–treatment-of-arrhythmia/ablation-for-arrhythmias.

[29] V. Fuster et al. “ACC/AHA/ESC 2006 Guidelines for the Management of Patients with Atrial Fibrillation–Executive Summary: A Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines and the European Society of Cardiology Committee for Practice Guidelines (Writing Committee to Revise the 2001 Guidelines for the Management of Patients with Atrial Fibrillation) Developed in Collaboration with the European Heart Rhythm Association and the Heart Rhythm Society”. In: *European Heart Journal* 27.16 (Aug. 2006), pp. 1979–2030. DOI: 10.1093/eurheartj/ehl176.

[30] *Ablation.* https://www.bhf.org.uk/informationsupport/treatments/ablation.

[31] H. Majeed and Y. Sattar. “Electrophysiologic Study Indications And Evaluation”. In: *StatPearls.* Treasure Island (FL): StatPearls Publishing, 2022.

[32] L. Muresan et al. “Recommendations for the Use of Electrophysiological Study: Update 2018”. In: *Hellenic journal of cardiology: HJC = Hellenike kardiologike epitheorese* 60.2 (2019), pp. 82–100. DOI: 10.1016/j.hjc.2018.09.002.

[33] D. P. Zipes et al. “Guidelines for Clinical Intracardiac Electrophysiological and Catheter Ablation Procedures: A Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines (Committee on Clinical Intracardiac Electrophysiologic and Catheter Ablation Procedures), Developed in Collaboration with the North American Society of Pacing and Electrophysiology”. In: *Journal of the American College of Cardiology* 26.2 (Aug. 1995), pp. 555–573. DOI: 10.1016/0735-1097(95)80037-H.

[34] D. P. Zipes et al. “2015 ACC/AHA/HRS Advanced Training Statement on Clinical Cardiac Electrophysiology (A Revision of the ACC/AHA 2006 Update of the Clinical Competence Statement on Invasive Electrophysiology Studies, Catheter Ablation, and Cardioversion)”. In: *Circulation: Arrhythmia and Electrophysiology* (2015). DOI: 10.1161/HAE.0000000000000014.

[35] W. Brady, ed. *Electrocardiogram in Clinical Medicine.* Hoboken, NJ: Wiley-Blackwell, 2020.

[36] F. M. Kusumoto. *ECG Interpretation: From Pathophysiology to Clinical Application.* Boston, MA: Springer US, 2009. DOI: 10.1007/978-0-387-88880-4.

[37] H. S. Oster, B. Taccardi, R. L. Lux, P. R. Ershler, and Y. Rudy. "Noninvasive Electrocardiographic Imaging: Reconstruction of Epicardial Potentials, Electrograms, and Isochrones and Localization of Single and Multiple Electrocardiac Events". In: *Circulation* 96.3 (Aug. 1997), pp. 1012–1024. DOI: 10.1161/01.cir.96.3.1012.

[38] Rudy Yoram. "Noninvasive Electrocardiographic Imaging of Arrhythmogenic Substrates in Humans". In: *Circulation Research* 112.5 (Mar. 2013), pp. 863–874. DOI: 10.1161/CIRCRESAHA.112.279315.

[39] C. D. Cantwell, Y. Mohamied, K. N. Tzortzis, S. Garasto, C. Houston, R. A. Chowdhury, F. S. Ng, A. A. Bharath, and N. S. Peters. "Rethinking Multiscale Cardiac Electrophysiology with Machine Learning and Predictive Modelling". In: *Computers in Biology and Medicine* 104 (Jan. 2019), pp. 339–351. DOI: 10.1016/j.compbiomed.2018.10.015.

[40] Y. Singh, D. Deepa, S.-Y. Wu, M. Friebe, J. Manuel R. S. Tavares, and W. Hu. "Cardiac Electrophysiology Studies Based on Image and Machine Learning". In: *Journal of Biomedical Engineering and Technology* 6.1 (Feb. 2018), pp. 1–6. DOI: 10.12691/jbet-6-1-1.

[41] S. Knecht et al. "Multicentre Evaluation of Non-Invasive Biatrial Mapping for Persistent Atrial Fibrillation Ablation: The AFACART Study". In: *EP Europace* 19.8 (Aug. 2017), pp. 1302–1309. DOI: 10.1093/europace/euw168.

[42] E. Pain. "How to Keep up with the Scientific Literature". In: *Science* (Nov. 2016). DOI: 10.1126/science.caredit.a1600159.

[43] P.-G. Majev, R. M. Vieira, A. Carollo, H. Liu, D. Stutz, A. Fahrenwaldt, N. Drummond, and M. P. P. S. Group 2020/2021. "PhDnet Report 2020". In: (Oct. 2021), p. 16704404. DOI: 10.17617/2.3344273.

[44] L. H. Olsthoorn, L. A. Heckmann, A. Filippi, R. M. Vieira, R. S. Varanasi, J. Lasser, F. Bäuerle, P. Zeis, R. Schulte-Sasse, and M. P. P. S. Group 2019/2020. "PhDnet Report 2019". In: (2020), pp. 11416677, 125050. DOI: 10.17617/2.3243876.

[45] X. Wang, S. Xu, L. Peng, Z. Wang, C. Wang, C. Zhang, and X. Wang. "Exploring Scientists' Working Timetable: Do Scientists Often Work Overtime?" In: *Journal of Informetrics* 6.4 (Oct. 2012), pp. 655–660. DOI: 10.1016/j.joi.2012.07.003.

[46] F. Spreckelsen, B. Rüchardt, J. Lebert, S. Luther, U. Parlitz, and A. Schlemmer. "Guidelines for a Standardized Filesystem Layout for Scientific Data". In: *Data* 5.2 (June 2020), p. 43. DOI: 10.3390/data5020043.

[47] E. D. Canale, G. R. Campbell, J. J. Smolich, and J. H. Campbell. *Cardiac Muscle*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986. DOI: 10.1007/978-3-642-50115-9.

[48] E. S. Richardson, A. J. Hill, N. D. Skadsberg, M. Ujhelyi, Y.-F. Xiao, and P. A. Iaizzo. "The Pericardium". In: *Handbook of Cardiac Anatomy, Physiology, and Devices*. Ed. by P. A. Iaizzo. Cham: Springer International Publishing, 2015, pp. 163–174. DOI: 10.1007/978-3-319-19464-6_9.

[49] T. G. Laske, M. Shrivastav, and P. A. Iaizzo. “The Cardiac Conduction System”. In: *Handbook of Cardiac Anatomy, Physiology, and Devices*. Ed. by P. A. Iaizzo. Cham: Springer International Publishing, 2015, pp. 215–233. DOI: 10.1007/978-3-319-19464-6_13.

[50] M. Pavelka and J. Roth. *Functional Ultrastructure*. Vienna: Springer Vienna, 2015. DOI: 10.1007/978-3-7091-1830-6.

[51] W. C. De Mello and M. J. Janse. *Heart Cell Coupling and Impulse Propagation in Health and Disease*. Vol. 12. Basic Science for the Cardiologist. Boston, MA: Springer US, 2002. DOI: 10.1007/978-1-4615-1155-7.

[52] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. *Mathematical Cardiac Electrophysiology*. Vol. 13. MS&A. Cham: Springer International Publishing, 2014. DOI: 10.1007/978-3-319-04801-7.

[53] H. A. Fozzard. “Heart: Excitation-Contraction Coupling”. In: (), p. 20.

[54] J. P. Keener and J. Sneyd. *Mathematical Physiology*. Interdisciplinary Applied Mathematics v. 8. New York: Springer, 1998.

[55] P. P. Sengupta, J. Korinek, M. Belohlavek, J. Narula, M. A. Vannan, A. Jahangir, and B. K. Khandheria. “Left Ventricular Structure and Function: Basic Science for Cardiac Imaging”. In: *Journal of the American College of Cardiology*. Focus Issue: Cardiac Imaging 48.10 (Nov. 2006), pp. 1988–2001. DOI: 10.1016/j.jacc.2006.08.030.

[56] M. Valderrábano. “Influence of Anisotropic Conduction Properties in the Propagation of the Cardiac Action Potential”. In: *Progress in biophysics and molecular biology* 94.1-2 (2007), pp. 144–168. DOI: 10.1016/j.pbiomolbio.2007.03.014.

[57] I. Kotadia, J. Whitaker, C. Roney, S. Niederer, M. O’Neill, M. Bishop, and M. Wright. “Anisotropic Cardiac Conduction”. In: *Arrhythmia & Electrophysiology Review* 9.4 (Dec. 2020), pp. 202–210. DOI: 10.15420/aer.2020.04.

[58] R. H. Clayton, O. Bernus, E. M. Cherry, H. Dierckx, F. H. Fenton, L. Mirabella, A. V. Panfilov, F. B. Sachse, G. Seemann, and H. Zhang. “Models of Cardiac Tissue Electrophysiology: Progress, Challenges and Open Questions”. In: *Progress in Biophysics and Molecular Biology*. Cardiac Physiome Project: Mathematical and Modelling Foundations 104.1 (Jan. 2011), pp. 22–48. DOI: 10.1016/j.pbiomolbio.2010.05.008.

[59] C. Mekkaoui, T. G. Reese, M. P. Jackowski, H. Bhat, and D. E. Sosnovik. “Diffusion MRI in the Heart”. In: *Nmr in Biomedicine* 30.3 (Mar. 2017), e3426. DOI: 10.1002/nbm.3426.

[60] R. A. Gray, A. M. Pertsov, and J. Jalife. “Spatial and Temporal Organization during Cardiac Fibrillation”. In: *Nature* 392.6671 (Mar. 1998), pp. 75–78. DOI: 10.1038/32164.

[61] R. A. Gray and J. Jalife. “Spiral Waves and the Heart”. In: *International Journal of Bifurcation and Chaos* 06.03 (Mar. 1996), pp. 415–435. DOI: 10.1142/S0218127496000163.

[62] K. H. Ten Tusscher, R. Hren, and A. V. Panfilov. “Organization of Ventricular Fibrillation in the Human Heart”. In: *Circulation Research* 100.12 (June 2007), e87–e101. DOI: 10.1161/CIRCRESAHA.107.150730.

[63] D. A. Hooks, K. A. Tomlinson, S. G. Marsden, I. J. LeGrice, B. H. Smaill, A. J. Pullan, and P. J. Hunter. “Cardiac Microstructure”. In: *Circulation Research* 91.4 (Aug. 2002), pp. 331–338. DOI: 10.1161/01.RES.0000031957.70034.89.

[64] D. B. Geselowitz. “On the Theory of the Electrocardiogram”. In: *Proceedings of the IEEE* 77.6 (June 1989), pp. 857–876. DOI: 10.1109/5.29327.

[65] D. B. Foster and D. B. Foster. *Twelve-Lead Electrocardiography: Theory and Interpretation.* 2nd ed. New York London: Springer, 2007.

[66] F. M. Kusumoto. *ECG Interpretation: From Pathophysiology to Clinical Application.* Second edition. Cham: Springer, 2020.

[67] C. Ramanathan, R. N. Ghanem, P. Jia, K. Ryu, and Y. Rudy. “Noninvasive Electrocardiographic Imaging for Cardiac Electrophysiology and Arrhythmia”. In: *Nature Medicine* 10.4 (Apr. 2004), pp. 422–428. DOI: 10.1038/nm1011.

[68] Y. Rudy and J. E. Burnes M.S. “Noninvasive Electrocardiographic Imaging”. In: *Annals of Noninvasive Electrocardiology* 4.3 (1999), pp. 340–359. DOI: 10.1111/j.1542-474X.1999.tb00220.x.

[69] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K.-A. Mardal, and A. Tveito. *Computing the Electrical Activity in the Heart.* Monographs in Computational Science and Engineering. Berlin Heidelberg: Springer-Verlag, 2006.

[70] J. Heijman, P. Erfanian Abdoust, N. Voigt, S. Nattel, and D. Dobrev. “Computational Models of Atrial Cellular Electrophysiology and Calcium Handling, and Their Role in Atrial Fibrillation”. In: *The Journal of Physiology* 594.3 (2016), pp. 537–553. DOI: 10.1113/JP271404.

[71] R. L. Winslow, S. Cortassa, B. O’Rourke, Y. L. Hashambhoy, J. J. Rice, and J. L. Greenstein. “Integrative Modeling of the Cardiac Ventricular Myocyte”. In: *WIREs Systems Biology and Medicine* 3.4 (2011), pp. 392–413. DOI: 10.1002/wsbm.122.

[72] W.-Y. I. Tseng, V. J. Wedeen, T. G. Reese, R. N. Smith, and E. F. Halpern. “Diffusion Tensor MRI of Myocardial Fibers and Sheets: Correspondence with Visible Cut-Face Texture”. In: *Journal of Magnetic Resonance Imaging* 17.1 (2003), pp. 31–42. DOI: 10.1002/jmri.10223.

[73] D. Braess. *Finite Elemente.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-34797-9.

[74] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Ed. by J. E. Marsden, L. Sirovich, and S. S. Antman. Vol. 15. Texts in Applied Mathematics. New York, NY: Springer New York, 2008. DOI: 10.1007/978-0-387-75934-0.

[75] H. P. Langtangen and A. Logg. *Solving PDEs in Python.* Cham: Springer International Publishing, 2016. DOI: 10.1007/978-3-319-52462-7.

[76] J. Cigánek and Z. Raida. “Temporal Discretization Schemes of Finite Element Methods: A Critical Comparison”. In: *2011 International Conference on Electromagnetics in Advanced Applications.* Sept. 2011, pp. 307–310. DOI: 10.1109/ICEAA.2011.6046306.

[77] C. Grossmann, H.-G. Roos, and M. Stynes. *Numerical Treatment of Partial Differential Equations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. DOI: 10.1007/978-3-540-71584-9.

[78] N. Beagrie. "Digital Curation for Science, Digital Libraries, and Individuals". In: *International Journal of Digital Curation* 1 (Nov. 2006), pp. 3–16. DOI: 10.2218/ijdc.v1i1.2.

[79] *What Is Digital Curation? | DCC*. https://www.dcc.ac.uk/about/digital-curation.

[80] M. Barisits et al. "Rucio: Scientific Data Management". In: *Computing and Software for Big Science* 3.1 (Aug. 2019), p. 11. DOI: 10.1007/s41781-019-0026-3.

[81] A. Lanati. *Quality Management in Scientific Research*. Cham: Springer International Publishing, 2018. DOI: 10.1007/978-3-319-76750-5.

[82] M. Baker. "1,500 Scientists Lift the Lid on Reproducibility". In: *Nature* 533.7604 (May 2016), pp. 452–454. DOI: 10.1038/533452a.

[83] F. S. Collins and L. A. Tabak. "Policy: NIH Plans to Enhance Reproducibility". In: *Nature* 505.7485 (Jan. 2014), pp. 612–613. DOI: 10.1038/505612a.

[84] J. Carlson, M. Fosmire, C. Miller, and M. S. Nelson. "Determining Data Information Literacy Needs: A Study of Students and Research Faculty". In: *portal: Libraries and the Academy* 11.2 (2011), pp. 629–657. DOI: 10.1353/pla.2011.0022.

[85] *About NSF - Overview | NSF - National Science Foundation*. https://www.nsf.gov/about/.

[86] National Academies of Sciences, Engineering, and Medicine (U.S.), National Academies of Sciences, Engineering, and Medicine (U.S.), National Academies of Sciences, Engineering, and Medicine (U.S.), and National Academies of Sciences, Engineering, and Medicine (U.S.), eds. *Reproducibility and Replicability in Science*. A Consensus Study Report of the National Academies of Sciences, Engineering, Medicine. Washington, DC: National Academies Press, 2019.

[87] T. Fitschen, A. Schlemmer, D. Hornung, H. tom Wörden, U. Parlitz, and S. Luther. "CaosDB—Research Data Management for Complex, Changing, and Automated Research Workflows". In: *Data* 4.2 (June 2019), p. 83. DOI: 10.3390/data4020083.

[88] B. Rüchardt. *Python Package: Good Scientific Practice*. https://gitlab.gwdg.de/bruecha/goodscientificpractice.

[89] *3D Slicer Image Computing Platform*. https://slicer.org/.

[90] A. Fedorov et al. "3D Slicer as an Image Computing Platform for the Quantitative Imaging Network". In: *Magnetic resonance imaging* 30.9 (Nov. 2012), pp. 1323–1341. DOI: 10.1016/j.mri.2012.05.001.

[91] T. Kapur et al. "Increasing the Impact of Medical Image Computing Using Community-Based Open-Access Hackathons: The NA-MIC and 3D Slicer Experience". In: *Medical Image Analysis*. 20th Anniversary of the Medical Image Analysis Journal (MedIA) 33 (Oct. 2016), pp. 176–180. DOI: 10.1016/j.media.2016.06.035.

[92] R. Kikinis, S. D. Pieper, and K. G. Vosburgh. "3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support". In: *Intraoperative Imaging and Image-Guided Therapy*. Ed. by F. A. Jolesz. New York, NY: Springer, 2014, pp. 277–289. DOI: 10.1007/978-1-4614-7657-3_19.

[93] *Image Segmentation — 3D Slicer Documentation*. https://slicer.readthedocs.io/en/latest/user_guide/image_segmentation.html.

[94] C. Pinter, A. Lasso, and G. Fichtinger. "Polymorph Segmentation Representation for Medical Image Computing". In: *Computer Methods and Programs in Biomedicine* 171 (Apr. 2019), pp. 19–26. DOI: 10.1016/j.cmpb.2019.02.011.

[95] D. Zukic, J. Vicory, M. McCormick, L. Wisse, G. Gerig, P. Yushkevich, and S. Aylward. "ND Morphological Contour Interpolation". In: *Insight J* (2016), pp. 1–8.

[96] L. Zhu, I. Kolesov, Y. Gao, R. Kikinis, and A. Tannenbaum. "An Effective Interactive Medical Image Segmentation Method Using Fast Growcut". In: *MICCAI Workshop on Interactive Medical Image Computing*. 2014.

[97] B. Weiß. "Segmentation, Registration and Virtual Reality Visualization of Multimodal Biomedical Images". Bachelor's Thesis. Göttingen: Georg August University of Göttingen, Feb. 2021.

[98] S. Wassing. "Finite Element Simulations of Cardiac Defibrillation Currents in Realistic Geometries". MA thesis. Göttingen: Georg August University of Göttingen, Jan. 2021.

[99] U. Ayachit. *The ParaView Guide: Updated for ParaView Version 4.3*. Ed. by L. Avila. Full color version. Clifton Park, NY: Kitware Inc, 2015.

[100] *ParaView*. https://www.paraview.org/.

[101] S. Pajevic and C. Pierpaoli. "Color Schemes to Represent the Orientation of Anisotropic Tissues from Diffusion Tensor Data: Application to White Matter Fiber Tract Mapping in the Human Brain". In: *Magnetic Resonance in Medicine* 42.3 (1999), pp. 526–540. DOI: 10.1002/(SICI)1522-2594(199909)42:3<526::AID-MRM15>3.0.CO;2-J.

[102] *Coordinate Systems - Slicer Wiki*. https://www.slicer.org/wiki/Coordinate_systems.

[103] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. "MeshLab: An Open-Source Mesh Processing Tool". In: (), p. 8.

[104] C. Geuzaine and J.-F. Remacle. "Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. DOI: 10.1002/nme.2579.

[105] O. Sorkine. "Laplacian Mesh Processing". In: (2005). DOI: 10.2312/egst.20051044.

[106] T. A. Burkhart, D. M. Andrews, and C. E. Dunning. "Finite Element Modeling Mesh Quality, Energy Balance and Validation Methods: A Review with Recommendations Associated with the Modeling of Bone Tissue". In: *Journal of Biomechanics* 46.9 (May 2013), pp. 1477–1488. DOI: 10.1016/j.jbiomech.2013.03.022.

[107] D. Alexander, C. Pierpaoli, P. Basser, and J. Gee. “Spatial Transformations of Diffusion Tensor Magnetic Resonance Images”. In: *IEEE Transactions on Medical Imaging* 20.11 (2001), pp. 1131–1139. DOI: 10.1109/42.963816.

[108] *Bioimpedance and Bioelectricity Basics - 3rd Edition.* https://www.elsevier.com/books/bioimpedance-and-bioelectricity-basics/grimnes/978-0-12-411470-8.

[109] C. G. Koay, J. D. Carew, A. L. Alexander, P. J. Basser, and M. E. Meyerand. “Investigation of Anomalous Estimates of Tensor-Derived Quantities in Diffusion Tensor Imaging”. In: *Magnetic Resonance in Medicine* 55.4 (2006), pp. 930–936. DOI: 10.1002/mrm.20832.

[110] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[111] C. Geuzaine. *Gmsh Tutorial – Stl Remeshing.* https://gitlab.onelab.info/gmsh/gmsh/blob/gmsh_4_10_1/tutorials/t13.geo. Dec. 2021.

[112] E. Marchandise, C. C. de Wiart, W. G. Vos, and C. Geuzaine. “High Quality Surface Remeshing Using Harmonic Maps. Part II: Surfaces with High Genus and of Large Aspect Ratio.” In: (2010), p. 23.

[113] E. Marchandise, J.-F. Remacle, and C. Geuzaine. “Optimal Parametrizations for Surface Remeshing”. In: *Engineering with Computers* 30.3 (July 2014), pp. 383–402. DOI: 10.1007/s00366-012-0309-3.

[114] J.-F. Remacle, C. Geuzaine, G. Compere, and E. Marchandise. “High Quality Surface Remeshing Using Harmonic Maps”. In: (2009), p. 24.

[115] J. Schöberl. “NETGEN An Advancing Front 2D/3D-mesh Generator Based on Abstract Rules”. In: *Computing and Visualization in Science* 1.1 (July 1997), pp. 41–52. DOI: 10.1007/s007910050004.

[116] R. J. Boulbes. *Troubleshooting Finite-Element Modeling with Abaqus: With Application in Structural Engineering Analysis.* Cham: Springer International Publishing, 2020. DOI: 10.1007/978-3-030-26740-7.

[117] E. Madenci and I. Guven. *The Finite Element Method and Applications in Engineering Using ANSYS®.* Boston, MA: Springer US, 2015. DOI: 10.1007/978-1-4899-7550-8.

[118] R. Stenger. “Locating Excitation Sites from Multichannel-ECG Signals”. Lab Rotation Report. Göttingen: Georg August University of Göttingen, May 2020.

[119] D. J. Griffiths. *Introduction to Electrodynamics.* Fourth edition. Cambridge, United Kingdom ; New York, NY: Cambridge University Press, 2018.

[120] M. J. Bishop and G. Plank. “Bidomain ECG Simulations Using an Augmented Monodomain Model for the Cardiac Source”. In: *IEEE Transactions on Biomedical Engineering* 58.8 (Aug. 2011), pp. 2297–2307. DOI: 10.1109/TBME.2011.2148718.

[121] “The Forward Euler Method”. In: *Practical Analysis in One Variable.* Ed. by D. Estep. Undergraduate Texts in Mathematics. New York, NY: Springer, 2002, pp. 583–604. DOI: 10.1007/0-387-22644-3_43.

[122] M. Potse, B. Dube, J. Richer, A. Vinet, and R. M. Gulrajani. “A Comparison of Monodomain and Bidomain Reaction-Diffusion Models for Action Potential Propagation in the Human Heart”. In: *IEEE Transactions on Biomedical Engineering* 53.12 (2006), pp. 2425–2435. DOI: 10.1109/TBME.2006.880875.

[123] J. R. Dormand and P. J. Prince. “A Family of Embedded Runge-Kutta Formulae”. In: *Journal of Computational and Applied Mathematics* 6.1 (Mar. 1980), pp. 19–26. DOI: 10.1016/0771-050X(80)90013-3.

[124] H. P. Langtangen and K.-A. Mardal. *Introduction to Numerical Methods for Variational Problems.* Vol. 21. Texts in Computational Science and Engineering. Cham: Springer International Publishing, 2019. DOI: 10.1007/978-3-030-23788-2.

[125] A. Logg, K.-A. Mardal, and G. Wells, eds. *Automated Solution of Differential Equations by the Finite Element Method.* Vol. 84. Lecture Notes in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-23099-8.

[126] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. “The FEniCS Project Version 1.5”. In: *Archive of Numerical Software* 3.100 (2015). DOI: 10.11588/ans.2015.100.20553.

[127] A. Logg and G. N. Wells. “DOLFIN: Automated Finite Element Computing”. In: *ACM Transactions on Mathematical Software* 37.2 (2010). DOI: 10.1145/1731022.1731030. arXiv: 1103.6248.

[128] A. Logg, G. N. Wells, and J. Hake. “DOLFIN: A C++/Python Finite Element Library”. In: *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering.* Ed. by A. Logg, K.-A. Mardal, and G. N. Wells. Springer, 2012. Chap. 10.

[129] S. Balay et al. *PETSc Web Page.* 2022.

[130] N. Sperelakis and T. Hoshiko. “Electrical Impedance of Cardiac Muscle”. In: *Circulation Research* 9.6 (Nov. 1961), pp. 1280–1283. DOI: 10.1161/01.RES.9.6.1280.

[131] S. Singh and S. Saha. “Electrical Properties of Bone. A Review.” In: *Clinical orthopaedics and related research* 186 (1984), pp. 249–271.

[132] C. D. Hansen and C. R. Johnson, eds. *The Visualization Handbook.* Amsterdam ; Boston: Elsevier-Butterworth Heinemann, 2005.

[133] R. Klepfer, C. Johnson, and R. Macleod. “The Effects of Inhomogeneities and Anisotropies on Electrocardiographic Fields: A 3-D Finite-Element Study”. In: *IEEE Transactions on Biomedical Engineering* 44.8 (Aug. 1997), pp. 706–719. DOI: 10.1109/10.605427.

[134] R. S. Zimmermann and U. Parlitz. “Observing Spatio-Temporal Dynamics of Excitable Media Using Reservoir Computing”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.4 (Apr. 2018), p. 043118. DOI: 10.1063/1.5022276.

[135] S. Berg, S. Luther, and U. Parlitz. “Synchronization Based System Identification of an Extended Excitable System”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 21.3 (Sept. 2011), p. 033104. DOI: 10.1063/1.3613921.

[136] J. Lebert and J. Christoph. “Synchronization-Based Reconstruction of Electromechanical Wave Dynamics in Elastic Excitable Media”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.9 (Sept. 2019), p. 093117. DOI: 10.1063/1.5101041.

[137] S. Herzog, R. S. Zimmermann, J. Abele, S. Luther, and U. Parlitz. “Reconstructing Complex Cardiac Excitation Waves From Incomplete Data Using Echo State Networks and Convolutional Autoencoders”. In: *Frontiers in Applied Mathematics and Statistics* 6 (2021).

[138] S. M. Lobodzinski. “ECG Instrumentation: Applicationand Design”. In: *Comprehensive Electrocardiology.* Ed. by P. W. Macfarlane, A. van Oosterom, O. Pahlm, P. Kligfield, M. Janse, and J. Camm. London: Springer London, 2010, pp. 427–480. DOI: 10.1007/978-1-84882-046-3_12.

[139] A. Mincholé and B. Rodriguez. “Artificial Intelligence for the Electrocardiogram”. In: *Nature Medicine* 25.1 (Jan. 2019), pp. 22–23. DOI: 10.1038/s41591-018-0306-1.

[140] A. Lyon, A. Mincholé, J. P. Martínez, P. Laguna, and B. Rodriguez. “Computational Techniques for ECG Analysis and Interpretation in Light of Their Contribution to Medical Advances”. In: *Journal of The Royal Society Interface* 15.138 (Jan. 2018), p. 20170821. DOI: 10.1098/rsif.2017.0821.

[141] R. C. Barr and A. van Oosterom. “Genesis of the Electrocardiogram”. In: *Comprehensive Electrocardiology.* Ed. by P. W. Macfarlane, A. van Oosterom, O. Pahlm, P. Kligfield, M. Janse, and J. Camm. London: Springer London, 2010, pp. 167–190. DOI: 10.1007/978-1-84882-046-3_5.

[142] B. M. Horáček. “Lead Theory”. In: *Comprehensive Electrocardiology.* Ed. by P. W. Macfarlane, A. van Oosterom, O. Pahlm, P. Kligfield, M. Janse, and J. Camm. London: Springer London, 2010, pp. 347–374. DOI: 10.1007/978-1-84882-046-3_10.

[143] H. C. Burger. “The Zero of Potential: A Persistent Error”. In: *American Heart Journal* 49.4 (Apr. 1955), pp. 581–586. DOI: 10.1016/0002-8703(55)90076-4.

[144] E. Frank. “General Theory of Heart-Vector Projection”. In: *Circulation Research* 2.3 (May 1954), pp. 258–270. DOI: 10.1161/01.RES.2.3.258.

[145] G. E. Dower, J. A. Osborne, and A. D. Moore. “MEASUREMENT OF THE ERROR IN WILSON’S CENTRAL TERMINAL: AN ACCURATE DEFINITION OF UNIPOLAR LEADS”. In: *Heart* 21.3 (July 1959), pp. 352–360. DOI: 10.1136/hrt.21.3.352.

[146] R. H. Bayley, E. W. Reynolds, C. L. Kinard, and J. F. Head. “The Zero of Potential of the Electric Field Produced by the Heart Beat: The Problem with Reference to Homogeneous Volume Conductors”. In: *Circulation Research* 2.1 (Jan. 1954), pp. 4–13. DOI: 10.1161/01.RES.2.1.4.

[147] R. H. Bayley and C. L. Kinard. “The Zero of Potential of the Electrical Field Produced by the Heart Beat”. In: *Circulation Research* 2.2 (Mar. 1954), pp. 104–111. DOI: 10.1161/01.RES.2.2.104.

[148] N. Miyamoto, Y. Shimizu, G. Nishiyama, S. Mashima, and Y. Okamoto. "On the Potential of the Wilson Central Terminal with Respect to an Ideal Reference for Unipolar Electrocardiography". In: *Journal of Electrocardiology* 28.4 (Oct. 1995), pp. 336–337. DOI: 10.1016/S0022-0736(05)80054-8.

[149] Y. Okamoto and S. Mashima. "The Zero Potential and Wilson's Central Terminal in Electrocardiography". In: *Bioelectrochemistry and Bioenergetics* 47.2 (Dec. 1998), pp. 291–295. DOI: 10.1016/S0302-4598(98)00201-3.

[150] H. Moeinzadeh, P. Bifulco, M. Cesarelli, A. L. McEwan, A. O'Loughlin, I. M. Shugman, J. C. Tapson, A. Thiagalingam, and G. D. Gargiulo. "Minimization of the Wilson's Central Terminal Voltage Potential via a Genetic Algorithm". In: *BMC Research Notes* 11.1 (Dec. 2018), p. 915. DOI: 10.1186/s13104-018-4017-y.

[151] G. D. Gargiulo. "True Unipolar ECG Machine for Wilson Central Terminal Measurements". In: *BioMed Research International* 2015 (2015), p. 586397. DOI: 10.1155/2015/586397.

[152] F. Fenton and A. Karma. "Vortex Dynamics in Three-Dimensional Continuous Myocardium with Fiber Rotation: Filament Instability and Fibrillation". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 8.1 (Mar. 1998), pp. 20–47. DOI: 10.1063/1.166311.

[153] S. Butterworth et al. "On the Theory of Filter Amplifiers". In: *Wireless Engineer* 7.6 (1930), pp. 536–541.

[154] A. Schlemmer, S. Berg, T. Lilienkamp, S. Luther, and U. Parlitz. "Spatiotemporal Permutation Entropy as a Measure for Complexity of Cardiac Arrhythmia". In: *Frontiers in Physics* 6 (2018).

[155] P. Pathmanathan, M. O. Bernabeu, R. Bordas, J. Cooper, A. Garny, J. M. Pitt-Francis, J. P. Whiteley, and D. J. Gavaghan. "A Numerical Guide to the Solution of the Bidomain Equations of Cardiac Electrophysiology". In: *Progress in Biophysics and Molecular Biology* 102.2–3 (June 2010), pp. 136–155. DOI: 10.1016/j.pbiomolbio.2010.05.006.

[156] F. Vadakkumpadan, L. J. Rantner, B. Tice, P. Boyle, A. J. Prassl, E. Vigmond, G. Plank, and N. Trayanova. "Image-Based Models of Cardiac Structure with Applications in Arrhythmia and Defibrillation Studies". In: *Journal of Electrocardiology* 42.2 (Mar. 2009), 157.e1–157.e10. DOI: 10.1016/j.jelectrocard.2008.12.003.

[157] G. D. Gargiulo, A. L. McEwan, P. Bifulco, M. Cesarelli, C. Jin, J. Tapson, A. Thiagalingam, and A. van Schaik. "Towards True Unipolar Bio-Potential Recording: A Preliminary Result for ECG". In: *Physiological Measurement* 34.1 (Dec. 2012), N1–N7. DOI: 10.1088/0967-3334/34/1/N1.

Appendix A

Appendix

A.1 Gmsh macros

Listing A.1 – *Macro to create bath in gmsh.*

```
Macro CreateBath
  // Create the bath with the coordinates specified by the
  // technical drawing from Laura Diaz.
  // Args:
  //  - height_bath: height of the bath that shall be assumed.
  // Defines:
  // - outer_boundary_bath_elements: Surface loop of surfaces that belong to the bath.
  //
  //
  heights_z = {0, height_bath};
  points[] = {};
  // make points
  For height_index In {0:1}
    coords[] = {};
    h_z = heights_z[height_index];
    p = {-92.02,  48.72,  h_z}; coords += p[];
    p = {-92.02, -48.72,  h_z}; coords += p[];
    // lower side
    p = {-48.72, -92.02, h_z}; coords += p[];
    p = {48.72, -92.02, h_z}; coords += p[];
    //  most right side
    p = { 92.02, -48.72,  h_z}; coords += p[];
    p = { 92.02,  48.72,  h_z}; coords += p[];
    // upper side
    p = {48.72, 92.02, h_z}; coords += p[];
    p = {-48.72, 92.02, h_z}; coords += p[];

    For ind In {0:#coords[]-3:3} // 3 coordinates per point -> steps of 3.
      x = coords[ind]; y = coords[ind + 1]; z = coords[ind + 2];
      p_new = newp; Point(p_new) = {x, y, z}; points += p_new;
    EndFor
```

```
EndFor

// make lines in each level; I assume more points per level than levels.
points_per_level = #points[]/2; // 2 levels
lines[] = {};
For p In {0:#points[]-1}
  start = points[p];
  If ((p+1)%points_per_level==0) // connect last in level with first in level.
    end = points[p+1-points_per_level];
  Else
    end = points[p+1];
  EndIf
  lt = newl; Line(lt)= {start, end}; lines += lt;
EndFor

// make lines connecting the leves
For p In {0:#points[]/2-1} // 2 levels
  start = points[p]; end = points[p+points_per_level];
  lt = newl; Line(lt)= {start, end}; lines += lt;
EndFor

// surfaces
lines_per_level = points_per_level; // since closed loop
surfaces[] = {};
// top and bottom
For l In {1:2} // 2 levels
  surface_lines = {};
  For line_index In {0:lines_per_level-1}
    line = lines[( (l-1)*lines_per_level ) + line_index];
    surface_lines += line;
  EndFor
  cvlp = newll ; Curve Loop(cvlp) = surface_lines[];
  s = news; Plane Surface(s) = {cvlp}; surfaces += s;
EndFor


// make surfaces side
For p In {0:points_per_level-1} // works only for 2 levels.
  If (p == points_per_level-1) // boundary always everything -8
    surface_lines[] = {
      lines[p], lines[p+points_per_level+1], -lines[p+points_per_level],
      -lines[p+2*points_per_level]
    };
  Else
    surface_lines[] = {
      lines[p], lines[p+2*points_per_level+1], -lines[p+points_per_level],
      -lines[p+2*points_per_level]
    };
  EndIf
  cvlp = newll ;Curve Loop(cvlp) = surface_lines[];
  s = news; Plane Surface(s) = {cvlp}; surfaces += s;
  EndFor
  outer_boundary_bath_elements[] = surfaces[];
Return
```

Listing A.2 – *Create a pair of dipol electrodes.*

```
Macro NormaliseVec
  // Normalises vector
  // Args:
  //   - VEC: vector to normalise
  // Returns
  //   - VEC_NORMALISED: normalised version of vector

  LENGTH = 0;
  For i In {0:#VEC[]-1}
    LENGTH += VEC[i]^2;
  EndFor
  LENGTH = Sqrt(LENGTH);

  VEC_NORMALISED [] = { };
  For i In {0:#VEC[]-1}
    VEC_NORMALISED += VEC[i] / LENGTH;
  EndFor
Return


Macro ScalarProduct
  // Calculate scalar product between two vectors
  // Args:
  //   - VEC1, VEC2: The vectors
  // Returns:
  //   - RES: the result

  RES = 0;
  For i In {0:#VEC1[]-1}
    RES += VEC1[i] * VEC2[i];
  EndFor
Return


Macro FindOnePerpendicularVector
  // Find a perpendicular vector to VEC
  // Args:
  //   - VEC: the vector (list with three coordinates)
  // Returns:
  //  - VEC_PERP: One random perpendicular vector to VEC
  // Idea: take scalar product with each coordinate axis. The largest angle
  // between VEC and the axis will give the smallest scalar product. This vector
  // will be taken as helper vector to calculate a perpendicular vector

  VEC1 [] = {1, 0, 0}; VEC2[] = VEC[];
  Call ScalarProduct;
  scalar_x = RES;

  VEC1 [] = {0, 1, 0}; VEC2[] = VEC[];
  Call ScalarProduct;
  scalar_y = RES;

  VEC1 [] = {0, 0, 1}; VEC2[] = VEC[];
  Call ScalarProduct;
  scalar_z = RES;

  // find closest-to perpendicular axis

  If (Fabs(scalar_x) >= Fabs(scalar_y))
    HELPER_VEC = {0, 1, 0};
    winner = scalar_y;
  Else
    HELPER_VEC = {1, 0, 0};
    winner = scalar_x;
  EndIf

  If (Fabs(winner) >= Fabs(scalar_z))
    HELPER_VEC = {0, 0, 1};
    winner = scalar_z;
  EndIf

  // create the perpendicular vector
  VEC_PERP [] = { };
  For i In {0:#VEC[]-1}
    VEC_PERP += HELPER_VEC[i] - winner * VEC[i];
  EndFor

  VEC[] = VEC_PERP[];
  Call NormaliseVec;
  VEC_PERP[] = VEC_NORMALISED[];
Return


Macro FindNormalVector
  // Find the normal vector to a surface defined by two vectors.
  // Args:
  //   - VEC1, VEC2: Two vectors as lists with three coordinates
  // Returns:
  //   - VEC_NORMAL: Normal vector to VEC1, VEC2
  // Note:
  //   - You have to make sure that VEC1 and VEC2 are not parallel.
  VEC_NORMAL = {};
  VEC_NORMAL += VEC1[1] * VEC2[2] - VEC1[2] * VEC2[1];
  VEC_NORMAL += VEC1[2] * VEC2[0] - VEC1[0] * VEC2[2];
  VEC_NORMAL += VEC1[0] * VEC2[1] - VEC1[1] * VEC2[0];

  VEC[] = VEC_NORMAL[];
  Call NormaliseVec;
  VEC_NORMAL[] = VEC_NORMALISED[];
Return


Macro CreateDipol
  // Create a dipol from two cylindres
  // Two types of arguments: 1) for the dipol, 2) for the electrodes
  // Args:
  //  - d_dipol: distance between dipol electrodes (surface electrode 1 to surface electrode 2)
  //  - centre_dipol: point between the two dipol electrodes (list with three coordinates)
  //  - direction_dipol: vector parallel to the vector
  //      connecting the two dipols electrode centres (list with three coordinates)
  //  - r_electrode: radius of the dipol electrodes
  //  - h_electrode: height of the dipol electrodes
  //  - normal_electrode: normal vector of the electrode (list with three coordinates)

  // Returns:
  //   - sf_elec1, sf_elec2: surfaces of elec1 and elec2

  // normalise direction
  VEC[] = direction_dipol[] ;
  Call NormaliseVec;
  direction_dipol[] = VEC_NORMALISED[];

  // normalise normal electrode
  VEC[] = normal_electrode[] ;
  Call NormaliseVec;
  normal_electrode[] = VEC_NORMALISED[];


  // centre electrodes
  // centre is dipol centre + half electrode distance +  radius electrode
  C1[] = {} ;
  C2[] = {} ;
  For i In {0:#centre_dipol[]-1}
    C1 += centre_dipol[i] + (d_dipol / 2  + r_electrode) * direction_dipol[i];
    C2 += centre_dipol[i] - (d_dipol / 2  + r_electrode) * direction_dipol[i];
  EndFor

  // create electrodes
  r_cylindre = r_electrode;
  h_cylindre = h_electrode;
  c_cylindre[] = C1[];
  normal[] = normal_electrode[];
  Call Cylindre;
  sf_elec1 [] = surfaces_cylindre[];

  c_cylindre[] = C2[];
  normal[] = normal_electrode[];
  Call Cylindre;
  sf_elec2 [] = surfaces_cylindre[];

Return


Macro Cylindre
  // Create a cylindre
  // Args:
  //  - r_cylindre: radius
  //  - h_cylindre: height
  //  - c_cylindre: centre of cylindre in height and radius (list with three coordinates)
  //  - normal: normal of cylindre (list with three coordinates)
  // Returns:
  //  - surfaces_cylindre

  // if normal of cylindre is not normalised, normalise it
  VEC[] = normal[];
  Call NormaliseVec;
  normal[] = VEC_NORMALISED[];

  centre_lower_surface[] = {};
  For i In {0:#c_cylindre[]-1}
    centre_lower_surface += c_cylindre[i] - 0.5 * normal[i] * h_cylindre ;
  EndFor

  // find two perpendicular vectors (V1, V2) in radial direction
  VEC [] = normal[];
  Call FindOnePerpendicularVector;
  V1 [] = VEC_PERP[];

  VEC1 = normal[];
  VEC2 = V1[];
  Call FindNormalVector;
  V2 [] = VEC_NORMAL [];

  // make a circle as base area
  x = centre_lower_surface[0]; y = centre_lower_surface[1]; z = centre_lower_surface[2];
  DELTA_R1 [] = {};
  DELTA_R2 [] = {};
  For i In {0:#V1[]-1}
    DELTA_R1 += r_cylindre * V1[i];
    DELTA_R2 += r_cylindre * V2[i];
  EndFor

  // circle
  p1 = newp; Point(p1) = {x, y  , z  };
  p2 = newp; Point(p2) = {x + DELTA_R1[0], y + DELTA_R1[1], z + DELTA_R1[2] };
  p3 = newp; Point(p3) = {x + DELTA_R2[0], y + DELTA_R2[1], z + DELTA_R2[2] };
  p4 = newp; Point(p4) = {x - DELTA_R1[0], y - DELTA_R1[1], z - DELTA_R1[2] };
  p5 = newp; Point(p5) = {x - DELTA_R2[0], y - DELTA_R2[1], z - DELTA_R2[2] };

  c1 = newl; Circle(c1) = {p2,p1,p3};
  c2 = newl; Circle(c2) = {p3,p1,p4};
  c3 = newl; Circle(c3) = {p4,p1,p5};
  c4 = newl; Circle(c4) = {p5,p1,p2};

  l1 = newll; Curve Loop(l1) = {c1, c2, c3, c4};
  cyl_base = news; Plane Surface(cyl_base) = {l1};

  // extrude to volume
  translation_vector[] = {};
  For i In {0:#normal[]-1}
    translation_vector += normal[i] * h_cylindre;
  EndFor

  out[] = Extrude{translation_vector[0], translation_vector[1],
    translation_vector[2]} {Surface{cyl_base};};

  // Remove volume created
  Delete {Volume{out[1]};}

  surfaces_cylindre [] = {cyl_base, out[0]};
  For i In {2:#out[] -1} // out[1] is the volume
    surfaces_cylindre += out[i];
  EndFor

Return
```

Listing A.3 – *Macro to create the electrode panel.*

```
// File contains:
// Macro for one electroe panel

Geometry.OldNewReg = 0;

Macro electrode
  // Create one electrode in two steps: first lower part (interface with panel)
  // then upper part (interface with bath), then connect both parts.
  lc_electrode = 3;

  // circle interface with panel
  p1 = newp; Point(p1) = {x, y  , z  } ; //,  lc_electrode} ;
  p2 = newp; Point(p2) = {x, y  , z+r} ; //,  lc_electrode} ;
  p3 = newp; Point(p3) = {x, y-r, z  } ; //,  lc_electrode} ;
  p4 = newp; Point(p4) = {x, y  , z-r} ; //,  lc_electrode} ;
  p5 = newp; Point(p5) = {x, y+r, z  } ; //,  lc_electrode} ;

  c1 = newl; Circle(c1) = {p2,p1,p3};
  c2 = newl; Circle(c2) = {p3,p1,p4};
  c3 = newl; Circle(c3) = {p4,p1,p5};
  c4 = newl; Circle(c4) = {p5,p1,p2};
```

```

// circle at interface with bath
d = 3; // depth electrode
p6 = newp; Point(p6) = {x + d, y  , z  } ; //,  lc_electrode} ;
p7 = newp; Point(p7) = {x + d, y  , z+r} ; //,  lc_electrode} ;
p8 = newp; Point(p8) = {x + d, y-r, z  } ; //,  lc_electrode} ;
p9 = newp; Point(p9) = {x + d, y  , z-r} ; //,  lc_electrode} ;
p10 = newp; Point(p10) = {x + d, y+r, z  } ; //,  lc_electrode} ;

c5 = newl; Circle(c5) = {p7,p6,p8};
c6 = newl; Circle(c6) = {p8,p6,p9};
c7 = newl; Circle(c7) = {p9,p6,p10};
c8 = newl; Circle(c8) = {p10,p6,p7};
For temp In {c1:c8}
  curves += temp;
EndFor

l1 = newll; Curve Loop(l1) = {c1, c2, c3, c4};
electrode_loops += l1 ; // electrode loops on panel
ls_panel = news; Plane Surface(ls_panel) = {l1};
surfaces += ls_panel;
surfaces_panel += ls_panel;

l1 = newll; Curve Loop(l1) = {c5, c6, c7, c8};
ls_bath = news; Plane Surface(ls_bath) = {l1};
surfaces += ls_bath;
interfaces_bath += ls_bath;

// connect upper circle and lower part.
l1 = newl; Line(l1) = {p2, p7};
l2 = newl; Line(l2) = {p3, p8};
l3 = newl; Line(l3) = {p4, p9};
l4 = newl; Line(l4) = {p5, p10};
For temp In {l1:l4}
  curves += temp;
EndFor

// make surfaces via lineloops
ll1 = newll; Curve Loop(ll1) = {l1, c5, -l2, -c1};
ll2 = newll; Curve Loop(ll2) = {l2, c6, -l3, -c2};
ll3 = newll; Curve Loop(ll3) = {l3, c7, -l4, -c3};
ll4 = newll; Curve Loop(ll4) = {l4, c8, -l1, -c4};

s5 = news; Surface(s5) = {ll1};
s6 = news; Surface(s6) = {ll2};
s7 = news; Surface(s7) = {ll3};
s8 = news; Surface(s8) = {ll4};
For temp In {s5:s8}
  surfaces += temp;
  interfaces_bath += temp;
EndFor

```

```
  surface_electrode = newsl;
  Surface Loop(surface_electrode) = {s5, s6, s7, s8, ls_bath, ls_panel};
  v_e = newv; Volume(v_e) = {surface_electrode};
  volumes += v_e;
  Physical Volume(electrode_counter) = {v_e};
  Printf("tag_ecg_electrode_%g=%g", electrode_counter, phys_tag) >> InfoFile;
  electrode_counter +=1;
  phys_tag += 1;
Return

Macro boxwithhole

  // create the box with a whole in it.
  /*//////// cylinder surface ///////////*/
  // lower
  p1 = newp; Point(p1) = {0,46,-80,lccylinder};
  p2 = newp; Point(p2) = {0,46+20,-80,lccylinder};
  p3 = newp; Point(p3) = {0,46,-80+20,lccylinder};
  p4 = newp; Point(p4) = {0,46-20,-80,lccylinder};
  p5 = newp; Point(p5) = {0,46,-80-20,lccylinder};

  l1 = newl; Circle(l1) = {p2,p1,p3};
  l2 = newl; Circle(l2) = {p3,p1,p4};
  l3 = newl; Circle(l3) = {p4,p1,p5};
  l4 = newl; Circle(l4) = {p5,p1,p2};

  For temp In {l1:l4}
    curves += temp;
  EndFor

  cl00 = newll; Curve Loop(cl00) = {l1, l2, l3, l4};
  // Printf("lower cylinder b: %f", cl00);
  lower_cylinder_boundary = cl00;

  // upper
  p6 = newp; Point(p6) = {5,46,-80,lccylinder};
  p7 = newp; Point(p7) = {5,46+20,-80,lccylinder};
  p8 = newp; Point(p8) = {5,46,-80+20,lccylinder};
  p9 = newp; Point(p9) = {5,46-20,-80,lccylinder};
  p10 = newp; Point(p10) = {5,46,-80-20,lccylinder};
  l5 = newl; Circle(l5) = {p7,p6,p8};
  l6 = newl; Circle(l6) = {p8,p6,p9};
  l7 = newl; Circle(l7) = {p9,p6,p10};
  l8 = newl; Circle(l8) = {p10,p6,p7};

  For temp In {l5:l8}
    curves += temp;
  EndFor

  cl0 = newll; Curve Loop(cl0) = {l5, l6, l7, l8};

  upper_cylinder_boundary = cl0;
```

```
// Printf("upper cylinder b: %f", upper_cylinder_boundary);

// curved surfaces in the cylinder
// make connections between cylinders
l9 = newl; Line(l9) = {p8, p3};
l10 = newl; Line(l10) = {p7, p2};
l11 = newl; Line(l11) = {p10, p5};
l12 = newl; Line(l12) = {p9, p4};
For temp In {l9:l12}
  curves += temp;
EndFor


//connect upper with lower and make surfaces.
// inner part of cylinder surface
cl1 = newll; Curve Loop(cl1) = {l9, -l1, -l10, l5};
cl2 = newll; Curve Loop(cl2) = {l10, -l4, -l11, l8};
cl3 = newll; Curve Loop(cl3) = {l7, l11, -l3, -l12};
cl4 = newll; Curve Loop(cl4) = {l12, -l2, -l9, l6};

s1 = news; Surface(s1) = {cl1};
s2 = news; Surface(s2) = {cl2};
s3 = news; Surface(s3) = {cl3};
s4 = news; Surface(s4) = {cl4};
For temp In {s1:s4}
  surfaces += temp;
  interfaces_bath += temp;
  surfaces_panel += temp;
EndFor

// make box
/*//////// panel surface ///////////*/
//lower part from top left counter clockwise
p11 = newp; Point(p11) = {0,0,0} ;//,lcbox};
p12 = newp; Point(p12) = {0,0,-160} ;//,lcbox};
p13 = newp; Point(p13) = {0,92,-160} ;//,lcbox};
p14 = newp; Point(p14) = {0,92,0} ;//,lcbox};
// connect points lower part
l13 = newl; Line(l13) = {p11, p12};
l14 = newl; Line(l14) = {p12, p13};
l15 = newl; Line(l15) = {p13,p14};
l16 = newl; Line(l16) = {p14,p11};
For temp In {l13:l16}
  curves += temp;
EndFor

cl19 = newll; Curve Loop (cl19) = {l13, l14, l15, l16};

// to create surface: cut off inner part of cylinder, inner part is c100
ns199 = news; Surface(ns199) = {cl19, cl00};
surfaces+= ns199;
interfaces_bath += ns199;
```

```
  surfaces_panel += ns199;

  //upper part
  p15 = newp; Point(p15) = {5,0,0} ;//,lcbox};
  p16 = newp; Point(p16) = {5,0,-160} ;//,lcbox};
  p17 = newp; Point(p17) = {5,92,-160} ;//,lcbox};
  p18 = newp; Point(p18) = {5,92,0} ;//,lcbox};
  // connect points upper part
  l17 = newl; Line(l17) = {p15, p16};
  l18 = newl; Line(l18) = {p16, p17};
  l19 = newl; Line(l19) = {p17, p18};
  l20 = newl; Line(l20) = {p18, p15};
  For temp In {l17:l20}
    curves += temp;
  EndFor

  // connect lines
  cl20 = newll; Curve Loop (cl20) = {l17, l18, l19, l20};

  bounding_upper_box = cl20;

  // connect parts
  l21 = newl; Line(l21) = {p11, p15};
  l22 = newl; Line(l22) = {p12, p16};
  l23 = newl; Line(l23) = {p13, p17};
  l24 = newl; Line(l24) = {p14, p18};
  For temp In {l21:l24}
    curves += temp;
  EndFor

  // make sides to surfaces
  cl21 = newll; Curve Loop (cl21) = {l21, l17, -l22, -l13};
  cl22 = newll; Curve Loop (cl22) = {l22, l18, -l23, -l14};
  cl23 = newll; Curve Loop (cl23) = {l23, l19, -l24, -l15};
  cl24 = newll; Curve Loop (cl24) = {l24, l20, -l21, -l16};

  s5 = news; Plane Surface(s5) = {cl21};
  s6 = news; Plane Surface(s6) = {cl22};
  s7 = news; Plane Surface(s7) = {cl23};
  s8 = news; Plane Surface(s8) = {cl24};
  For temp In {s5:s8}
    surfaces+= temp;
    interfaces_bath += temp;
    surfaces_panel += temp;
  EndFor

Return


Macro electrodepanel
  /// start making the geometry
  // meshparams
// gathering geometries
// Args: [incomplete]
// - InfoFile: Path to file where the info should be go to.



points [] = {}; // gathers all points used
curves [] = {}; // gathers all curves used
surfaces [] = {}; // gathers all surfaces used
surfaces_panel [] = {};
volumes [] = {};
interfaces_bath[] = {};
start_points = newp;

points += start_points;
lcbox = 10;
lccylinder = 6;

// list to gather curves on electrodes.
electrode_loops [] = { };

// box surfaces gathers all surfaces of the panel:
// electrode surfaces on panel and the box surfaces the electrodes
// cut out (so in total all box surfaces).

Call boxwithhole;

/// now electrodes
r = 2.5;
// positions of center
y_poses[] = {86, 66, 46, 26, 6};
z_poses[] = {32, 48, 64, 80, 96, 112, 128, 144};

// (y, z)
//exclude_combinations = [(26, 80), (46, 96), (46, 80), (46, 64), (66, 80)]

x = 5;
num_y = #y_poses[];
num_z = #z_poses[];
t=0;
For indy In {0:num_y-1}
  For indz In {0:num_z-1}
    /// check if this has to be skipped
    y = y_poses[indy];
    z = -z_poses[indz];
    // achtung, z hat ein minuszeichen, deshalb aufpassen.
    If (!((Fabs(y-26)<0.01 && Fabs(z+80) < 0.01) || (Fabs(y-46)<0.01 && Fabs(z+96) < 0.01) || (Fabs(y-46)<0.01 && Fabs(z+80) < 0.01) || (Fabs(y-46)<0.01 && Fabs(z+64) < 0.01) || (Fabs(y-66)<0.01 && Fabs(z+80) < 0.01)))
       Call electrode;
       t += 1;
     EndIf
    EndFor
  EndFor

  // surface upper
  // without electrode loops
  snum = news; Surface(snum) = {bounding_upper_box, upper_cylinder_boundary, electrode_loops[]};
  interfaces_bath += snum;
  surfaces_panel += snum;
  surfaces += snum;
  end_points = newp;
  points += end_points -1;


  // Printf("range points '%g' and '%g'", points[0], points[1]);
  the_num = newsl; Surface Loop(the_num) = {interfaces_bath[]};
  surface_loops_interface_bath += the_num;
  // Printf("surface loop number: '%g'", the_num);
  points[] = {points[0]:points[1]};

  // Printf("Len points, curves, surfaces: '%g', '%g', '%g'", #this_box_points[], #this_box_curves[], #this_box_surfaces[]);

Return
```

Listing A.4 – *Connected Surface Detection.*

```
// Find connected surfaces

/* note:
  The approach used below cannot be applied to surfaces of seperate volumes that touch each other. (For example if we would split the heart muscle in the different parts which boundaries touch each other). For this usecase one would have to use a different approach, which I cannot come up with write now.
*/

surface_pool[] = surfaces[];
next_surfaces_to_check[] = {};
surface_loops[] = {};
srfcs_forming_this_clstr[] = {};
first_srfc = surface_pool[0]; surface_pool -= first_srfc;
not_processed_connected_srfcs[] = {first_srfc};
srfcs_forming_this_clstr += first_srfc;

For all_surfaces_processed In {0:1}
   // now: find neighbours of one surface
   // draw first surface
   If (#not_processed_connected_srfcs[] == 0)
     // this surface cluster is done, reset for next one
     srfcs_forming_this_clstr[] = {};
     current_srfc = surface_pool[0]; surface_pool -= current_srfc;
     srfcs_forming_this_clstr += current_srfc;
  Else
    current_srfc = not_processed_connected_srfcs[0];
    not_processed_connected_srfcs -= current_srfc;
  EndIf
  // check to which not yet assigned to a surface cluster surface this is connected
  bndry_srfc = Boundary{Surface{current_srfc};};
  For os In {0:#surface_pool[]-1}
    bndry_srfc_temp[] = bndry_srfc[];
    srfc = surface_pool[os];
    bndry_srfc_temp -= Boundary{Surface{srfc};};
    num_common_bndry_elements = #bndry_srfc[] - #bndry_srfc_temp[];
    If (num_common_bndry_elements!=0)
      srfcs_forming_this_clstr += srfc;
      not_processed_connected_srfcs += srfc;
    EndIf
  EndFor
  surface_pool -= srfcs_forming_this_clstr[];

  If (#not_processed_connected_srfcs[] == 0)
    // this surface cluster is done, create loop
    surface_loop = newsl; Surface Loop(surface_loop) = {srfcs_forming_this_clstr[]};
    surface_loops += surface_loop;
  EndIf


  If ( #surface_pool[] + #not_processed_connected_srfcs[] == 0)
    all_surfaces_processed = 1;
  Else
    all_surfaces_processed = 0;
  EndIf
EndFor
```

Listing A.5 – *Macro create shock electrode.*

```
Macro electrode
  // input:
  // anker: one corner of the electrode (x,y,z)
  // side: (x,y,z) from anker the side
  // down: (x,y,z) from ankor down
  // normal: (x, y, z) direction and length of extrusion

  //// Path to go: (Instructions for me while coding)
  // in total: 8 points!
  // I will do: create plan, then extrude and get boundary by volume
  // first: create surface from points and curve loop,
  // then : extrude to volume

  // make points
  as[] = {};
  ad[] = {};
  asd[] = {};
```

```

For i In {0:2}
  as[] += anker[i] + side[i];
  ad[] += anker[i] + down[i];
  asd[] += anker[i] + side[i] + down[i];
EndFor

coords[] = {};
coords += anker[];
coords += as[];
coords += asd[];
coords += ad[];
// next layer
For i In {0:#coords[]-3:3}
  coords += coords[i] + normal[0];
  coords += coords[i+1] + normal[1];
  coords += coords[i+2] + normal[2];
EndFor

points[] = {};
For ind In {0:#coords[]-3:3} // 3 coordinates per point -> steps of 3.
  x = coords[ind]; y = coords[ind + 1]; z = coords[ind + 2];
  p_new = newp; Point(p_new) = {x, y, z}; points += p_new;
EndFor

//Panker = newp; Point(Panker) = {anker[0], anker[1], anker[2]}; points += Panker;
//Pas = newp; Point(Pas) = {as[0], as[1], as[2]}; points += Pas;
//Pad = newp; Point(Pad) = {ad[0], ad[1], ad[2]}; points += Pad;
//Pasd = newp; Point(Pasd) = {asd[0], asd[1], asd[2]}; points += Pasd;

// make lines in each level; I assume more points per level than levels.
points_per_level = #points[]/2; // 2 levels
lines[] = {};
For p In {0:#points[]-1}
  start = points[p];
  If ((p+1)%points_per_level==0) // connect last in level with first in level.
    end = points[p+1-points_per_level];
  Else
    end = points[p+1];
  EndIf
  lt = newl; Line(lt)= {start, end}; lines += lt;
EndFor

// make lines connecting the levels
For p In {0:#points[]/2-1} // 2 levels
  start = points[p]; end = points[p+points_per_level];
  lt = newl; Line(lt)= {start, end}; lines += lt;
EndFor

// surfaces
```

```
  lines_per_level = points_per_level; // since closed loop
  surfaces[] = {};
  // top and bottom
  For l In {1:2} // 2 levels
    surface_lines = {};
    For line_index In {0:lines_per_level-1}
      line = lines[( (l-1)*lines_per_level ) + line_index];
      surface_lines += line;
    EndFor
    cvlp = newll ; Curve Loop(cvlp) = surface_lines[];
    s = news; Plane Surface(s) = {cvlp}; surfaces += s;
  EndFor


  // make surfaces side
  For p In {0:points_per_level-1} // works only for 2 levels.
    If (p == points_per_level-1) // boundary always everything -8
      surface_lines[] = {
        lines[p], lines[p+points_per_level+1], -lines[p+points_per_level],
        -lines[p+2*points_per_level]
      };
    Else
      surface_lines[] = {
        lines[p], lines[p+2*points_per_level+1], -lines[p+points_per_level],
        -lines[p+2*points_per_level]
      };
    EndIf
    cvlp = newll ;Curve Loop(cvlp) = surface_lines[];
    s = news; Plane Surface(s) = {cvlp}; surfaces += s;
  EndFor

  bndry_elements[] = surfaces[];

Return
```

Listing A.6 – *Example usage of the Parameter storer class.*

```python
from ParameterStorage import ParameterStorer

ps = ParameterStorer()
# Sets the variable value to 17 and stores this information in the parameterstorer class
var = ps("Var_name", 17)
ps.write("/path/to/output/folder", filename="parameters.csv")
```

Acknowledgements

First of all I'll have to thank my supervisor Ulrich Parlitz and my group leader Stefan Luther for giving me the opportunity to do the PhD in the group. In German the PhD supervisor is called „Doktorvater“ (PhD father) and so I have to be Ulrichs Doktorsohn — PhD son. And I must confess, continuing with this analogy, I was a PhD son in puberty with all the emotions that go with it. I want to thank him that he went through all these phases that he probably knew from former PhD students. Also, I am especially grateful for his feedback at the end in my writing phase.

I want to thank them for shielding us, the group, from all the trouble that happen in a high-hierarchy environment like universities are and our Max Planck Institute is and for letting me experience what it means to be a scientist.

I want to thank my thesis advisory committee members Florentin Wörgötter and Andreas Neef for giving me always the opportunity to ask them and for being much more convinced in my work than I was. Sometimes, when one is trapped under the pile of work one only sees the huge amount of things that were not done, not was actually had been achieved. I also thank them for listening to my concerns. This was very helpful for me in the private sessions of the TAC meetings.

I want to take the Max Planck Society for the effort and opportunities they make and give to help the employees in their personal development. I learned a lot doing the courses of the Planck Academy and I enjoy being connected with Max Planck students from other universitites and Max Planck Institutes.

I want to thank Simon Wassing, a master student I supervised, for working together with me with a nonlinear increase in output! It really was a pleasure to have such a fruitful work time together.

Alexander Schlemmer was very important to me all over the PhD but especially in its last phase for scientific discussions and for discussions about the science that was important to me. He listened, gave scientific input and tried to understand together with me what my current scientific tasks were and what they were good for. He, Johanna and their Miriam provided a lot of support in the writing phase and in handling the difficulties in the scientific environment.

I cannot tell how grateful I am to Florian Spreckelsen for proofreading my thesis, giving input asking the right questions, not getting tired of highlighting the spelling mistakes and sentences that were revised until they didn't make any sense anymore.

Also I want to mention Sayedeh who after her thesis learned that things are okay and who always listens and can give the feeling of things just being right. Then I want to thank my colleague? friend! Laura Díaz for all the chats discussing our work

[conditions], private stuff or whatsoever. Similarly I want to thank Justine Wolter who is the best office mate one can imagine — no matter whether we talk, we throw our tiny soft office ball (or rubber ducks) at each other, whether we share fruits or cake. Whether we discuss our current tasks or whether she stands my great jokes. This was amazing!

This thesis would not have been possible without the support of many of the greatest people one can imagine. When I wrote the dedication at the beginning of the thesis I wrote to ‘my families’ and besides the fantastic members of my actual family I meant people that have become like family to me. This are Bérénice and Alexander with who I can have the longest evenings and wouldn’t be tired of then have the longest day just afterwards. I also mean my flat mates, first of all Hafssa and Lars, and later Vanessa: Having met you was certainly worth all the troubles that doing a PhD brings with it. And there is Antonio who I would adopt as a brother if this was possible.

Speaking of brothers, I have to mention my brother Anatol and my chosen sister Kesso. It is so nice to have you close and the world would be a worse place without you.

Then there are more people that span a net below me while I balance on the rope that is life. I was never afraid because I know there are people like Célia and Jamila, like Konrad or Raphael. There is also Carlotta who propped me up when I was so bad I do not wish anyone to be in a comparable situation and Janna who also was a light in these times. And special thanks to Filippo, my best colleague, who made my start in this working group — back in the time as a Master student — so enjoyable! And to Selene, with both of them I’ve spent (and plan to spend) so good times!

Last but not least I want to mention Mona. Even though I didn’t want to be in a relationship with you anymore in 2020, I am so incredibly thankful for the 8 years before and I cannot tell how it breaks my heart over and over again knowing that since March of this year (2022) we are no longer living in the same shared reality, that we cannot watch by coincidence the same TV program and that we will never have the opportunity to sit together and be sad about what has happened to us while appreciating what we had.