# Modeling the Electrical Activity in the Ventricles with Focus on the Purkinje-Myocardial Coupling

by

KAROLINE KALLEBERG

***THESIS***

*for the degree of*

***MASTER OF SCIENCE***

*(Master i Anvendt matematikk og mekanikk)*


*Faculty of Mathematics and Natural Sciences*
*University of Oslo*

*May 2015*

*Det matematisk- naturvitenskapelige fakultet*
*Universitetet i Oslo*

# Acknowledgements

Working on this thesis has had its ups and downs, as I am sure it has for everyone writing a thesis, but throughout the three semesters that I have been working on this thesis I have always felt very lucky to have Joakim Sundnes as my supervisor. All this time he has been kind and friendly, and when you feel really stupid that makes a world of difference. He has also helped me a lot with all the problems emerging on the way, being available for meeting up and answering my emails fast. I am very grateful that you have made this time easier than I am sure it would have been otherwise, Joakim. Thank you.

I want to thank the people at Simula Research Laboratory for an open and frendly working environment. In particular I would like to thank Glenn Terje Lines and Siri Kallhovd for providing code to generate the Purkinje network and defining the myocardium, Martin Sandve Alnæs for help on coupling the Purkinje network and the myocardial tissue and Pan Li, who had the idea for this thesis and provided the C code for the PRd cell model.

Thank you Bjarne for reading through this whole thesis and giving lots of useful comments, for technical support, many helpful discussions, putting up with my frustration, making dinner and vacuum cleaning more than your part and generally being there for me.

Thank you Karoline for five years of studies together. You have been a great support throughout this whole time, being my friend and making me see new things during our discussions.

I would like to thank Torstein for reading through my thesis and giving useful comments.

Finally, a general thank you to my Mum and Dad and my friends Mailis and Celine for always being there for me.

# Notation

## Abbreviations

- AV node - atrioventricular node
- CAMKII - $Ca^{2+}$/calmodulin-dependent protein kinase II
- CMDN - calmodulin
- CSQN - calsequestrin
- CSR - corbular sarcoplasmic reticulum compartment
- FHN - FitzHugh-Nagumo
- JSR - junctional sarcoplasmic reticulum compartment
- Myo - bulk myoplasma compartment
- MYOPLASM - myoplasma compartment
- NSR - network sarcoplasmic reticulum compartment
- ODE - ordinary differential equation
- PDE - partial differential equation
- PCS - peripheral coupling subspace compartment
- PMJ - Purkinje-myocardial junction
- PRd - Purkinje cell model by Li and Rudy [17]
- SA node - sinoatrial node
- SCD - sudden cardiac death
- SR - sarcoplasmic reticulum
- SS(CaL) - $I_{CaL}$ subspace compartment
- SSL - subsarcolemmal compartment
- SS(SR) - sarcoplasmic reticulum subspace compartment
- TRPN - troponin

# Symbols

- $[X]$, where $X$ is an element - ionic concentration [mmol/L]
- $\alpha, \beta$ - rate constants [1/ms]
- $\sigma$ - conductivity [mS/cm]
- $\sigma_{PMJ}$ - conductance at the PMJs [mS]
- $\chi$ - amount of membrane surface area per unit volume of tissue [1/cm]
- $A_{cap}$ - capacitive membrane area [cm$^2$]
- $B$ - magnetic field [N/($10^{10}$ $\mu$A cm)] = [0.01 T]
- $c$ - ionic concentration [mmol/L]
- $C_m$ - capacitance per area [$\mu$F/cm$^2$]
- $D$ - diffusivity, diffusion coefficient [cm$^2$/ms]
- $D_{PMJ}$ - "diffusion coefficient" at the PMJs [cm$^3$/ms]
- $E$ - electric field [mV/cm]
- $F$ - Faraday's constant [C/mol]
- $G_{max}$ - conductance per area with $1/C_m$ incorporated [mS/$\mu$F]
- $I$ - current per area [$\mu$A/cm$^2$], or [$\mu$A/$\mu$F] if $1/C_m$ is incorporated
- $J$ - flux defined with respect to a reference volume [mmol/(L ms)]
- $M$ - conductivity [mS/cm]
- $P$ - membrane permeability to an ion [cm/s]
- $q$ - charge per area [nC/cm$^2$]
- $R$ - ideal gas constant [J/(kmol K)]
- $T$ - temperature [K]
- $t$ - time [ms]
- $V$ - potential [mV]
- $V_X$, where $X$ is a compartment - compartment volume [$\mu$L]
- $z$ - valence of an ion [1]

# Contents

**Acknowledgements** iii

**Notation** v
Abbreviations . . . v
Symbols . . . vi

**1 Introduction** 1

**2 Physiological Background** 3
2.1 Motivation: Sudden Cardiac Death . . . 3
2.2 The Heart . . . 5
2.2.1 The Circulatory System . . . 5
2.2.2 The Conduction System . . . 6
2.3 The Ventricles . . . 6
2.4 Cell Physiology . . . 7
2.4.1 Anatomy . . . 8
2.4.2 The Cell Membrane . . . 8
2.4.3 Action Potential and Ionic Currents . . . 10
2.4.4 Excitation-Contraction Coupling . . . 13
2.4.5 Cell Connectivity . . . 14
2.5 What to Model . . . 14

**3 Mathematical Models** 15
3.1 Single Cell Models . . . 15
3.1.1 A Cubic Polynomial . . . 17
3.1.2 The FHN Model . . . 18
3.1.3 The PRd Model . . . 20
3.1.4 The Decker et al. Model . . . 33
3.1.5 Comparing the PRd and the Decker et al. Models . . . 38
3.2 Cardiac Tissue Model . . . 40
3.2.1 The Monodomain Equation . . . 40

**4 Numerical Methods** **43**
4.1 Time Discretization . . . 43
4.1.1 Operator Splitting . . . 43
4.1.2 The Finite Difference Method . . . 45
4.2 Spatial Discretization . . . 46
4.2.1 Weak Form . . . 47
4.2.2 One Equation on the Whole Purkinje Domain . . . 49
4.3 Coupling the Purkinje Network and the Myocardial Tissue . . . 50
4.3.1 Coupling Through Boundary Conditions . . . 51
4.3.2 Coupling Through a Stimulus Current . . . 53

**5 Implementation** **57**
5.1 Programming Packages and Received Code . . . 58
5.1.1 FEniCS . . . 58
5.1.2 Gotran . . . 58
5.1.3 Goss . . . 59
5.1.4 Plotting Packages . . . 59
5.1.5 PRd Cell Model and Monodomain Solver . . . 60
5.1.6 Code to Define the Myocardium and Generate a Purkinje Network . . . 60
5.2 Meshes . . . 62
5.2.1 The Purkinje Network . . . 62
5.2.2 The Myocardial Tissue . . . 66
5.3 Cell Models . . . 67
5.3.1 A Single Cell . . . 68
5.3.2 Cell Models in Space: Goss-Solver . . . 71
5.4 Tissue Model . . . 72
5.4.1 The Linear PDE . . . 72
5.5 Coupling the Purkinje Network and the Myocardial Tissue . . . 73
5.5.1 One-Way Coupling . . . 73
5.5.2 Two-Way Coupling . . . 75

**6 Results** **81**
6.1 Tissue Model . . . 81
6.1.1 First Test Case: The Bistable Equation . . . 81
6.1.2 Validating the Midpoint Method: FHN Cell Model . . . 82
6.1.3 Validating the Goss-Solver: FHN Cell Model . . . 84
6.1.4 Testing the Modified FHN Cell Model . . . 86
6.1.5 Testing the PRd Cell Model . . . 87
6.1.6 Signal Propagation in the Opposite Direction . . . 92
6.1.7 Realistic Tree . . . 93

6.2 Coupling the Purkinje Network and the Myocardial Tissue . . 96
6.2.1 One-Way Coupling . . . . . 96
6.2.2 Two-Way Coupling . . . . . 99
6.3 Left Bundle Branch Block . . . . . 107

**7 Discussion and Concluding Remarks** **111**
7.1 The Purkinje Network . . . . . 111
7.2 The Myocardial Tissue . . . . . 114
7.3 Coupling of the Purkinje Network and the Myocardial Tissue . 115
7.4 Resolution . . . . . 119
7.5 Numerical Methods . . . . . 119
7.6 Concluding Remarks . . . . . 120

**Bibliography** **123**

# Chapter 1

# Introduction

> "The single most important cause of death in the adult population of the industrialized world is SCD [sudden cardiac death] due to coronary disease."

This is stated in a task force on sudden cardiac death (SCD) of the European Society of Cardiology by Priori et al. [21, p. 1377]. At first this might sound like an exaggeration because although too many die prematurily very many people simply seem to die of old age, falling asleep never to wake up again. But "old age" is not a scientifically accepted cause of death, there is always a specific reason why the body stops working, and this very often has to do with the heart. According to Mehra [20, p. S118] there are almost 17 million deaths annually resulting from cardiovascular disease, making up 30% of the global deaths. In turn 40-50% of these are assumed to be SCDs, which amounts to between 6.8 million and 8.5 million people, and 80% of these again are caused by ventricular tachyarrhythmias.

Medical problems are inverse: we see only the symptoms, not the cause. To be able to treat a case in the best possible way, starting focused treatment fast, it is crucial to know the cause. Computer models help researchers attain this knowledge.

With a mathematical model of the case in study, it is possible to simulate the time development of the system on a computer. Varying model parameters, the role of the different parts of the model, and hopefully also of the real physiological system, can be investigated. If the computer models give strong indications that a specific symptom has a specific cause, this can be further tested in clinical experiments, in the end leading to more targeted and better treatment. Computer models have the superior advantage that once they are made, they can be modified to model many different cases, and they can be run as many times as desired for as long a time as deemed

suitable.

The purpose of this thesis is to model the electrical activity in the ventricles with the hope that this may contribute to the study of underlying mechanisms of arrhythmia that cause SCD. Ventricular arrhythmias make up the greatest part of the SCDs, so this is why the focus has been on the ventricles.

The implemented models show how electrical signals spread through the Purkinje network, and from there to the rest of the ventricular muscle tissue. For Purkinje cells the PRd cell model by Pan Li and Yoram Rudy from 2011 is used [17], while the Decker et al. cell model by Decker et al. from 2009 [15] is used for the cells in the myocardium. These models consist of ordinary differential equations (ODEs) governing a single cell, so to model multiple cells, the cell models were coupled to the monodomain equation, which is a partial differential equation (PDE). Using operator splitting on the monodomain equation with a cell model coupled to it, this system was split into a coupled system of nonlinear ODEs and a linear PDE. FEniCS was used to solve the PDE, while the system of ODEs were solved with goss.

As the Purkinje cell model is relatively new, the hope is that it can give some new insights about the electrical activity in the ventricles. While earlier models have adopted intracellular calcium dynamics from the cell models of ventricular myocytes, the PRd model has a newly developed mathematical model of the Purkinje cell that represents its unique electrophysiological and calcium handling properties.

The outline of the thesis is as follows: the physiological background for understanding the electrical activity in the heart and how this controls the blood flow pumped through the heart is given in Chapter 2. In Chapter 3 models for the electrical signal in single cells and tissue are described. Next is Chapter 4, where the numerical methods used on the model equations to get them into a form that can be solved computationally are presented, as well as how the coupling of the specialized conduction system of the heart and the heart muscle is modeled. Following this, the implementation of several models is explained in Chapter 5, validation and test cases being presented in Chapter 6, as well as a few results for the final implementation. Finally, the limitations and qualities of the models are discussed in Chapter 7, together with some concluding remarks.

# Chapter 2

# Physiological Background

As stated in the introduction, this thesis aims to simulate the electrical activity in the ventricles in the heart. An overview of the workings of the heart is presented, followed by a quick view on the physiology of a cell, but first comes a closer look on the term "sudden cardiac death", as this is the motivation for the project.

## 2.1 Motivation: Sudden Cardiac Death

In the aforementioned task force by Priori et al. [21, p. 1377], sudden cardiac death (SCD) is defined as:

> "Natural death due to cardiac causes, heralded by abrupt loss of consciousness within one hour of the onset of acute symptoms; preexisting heart disease may have been known to be present, but the time and mode of death are unexpected."

This is one definition among a variety. As discussed in the paper, there is no standard scientific definition of the term. The term "sudden cardiac death" has been in use for several centuries, and in some sense it is self-explanatory: death happening suddenly from causes that have to do with the heart. But how fast is "sudden"? In the mentioned paper and many others it is defined to be within one hour from the first symptoms. Earlier the time frame was 24 hours, increasing the number of people included in the definition. Another complicating factor is that many people pass away without any witnesses, leaving at least the time frame of the symptoms an open question, and in some cases also the actual cause of death. The lack of one standardized definition leads to the number of people counted as dead

from “sudden cardiac death” varying somewhat according to the definition, making comparisons of different studies problematic.

The definition is also quite wide as to the cause of death, it only has to be “cardiac”. Thus “sudden cardiac death” consists of several ways a heart can fail. Here again the fact that many people pass away unwitnessed - and more again pass away unmonitored - is complicating, because then the heart rhythm is not measured. In the introduction it was stated that there are almost 17 million deaths annually resulting from cardiovascular disease, making up 30% of the global deaths. In turn 40-50% are assumed to be SCDs, which amounts to between 6.8 million and 8.5 million people. Out of these, 80% are caused by ventricular tachyarrhythmias according to Mehra [20, p. S118]. The task force by Priori et al. [21, p. 1377] states that: “The first recorded rhythm in patients presenting with a sudden cardiovascular collapse is ventricular fibrillation (VF) in 75–80%”. These numbers and causes are similar, but not identical, showing that the results in this area are not global truths, but varying due to many factors. Still, ventricular arrhythmia is clearly the most common cause of SCD. During arrhythmia, the heart is not stimulated according to its normal pattern. This can make parts of the heart contract out of sync with the rest of the heart, leading to irregular contraction of the heart muscle. This again causes ineffective pumping of the blood from the heart to the body, which in the worst case is lethal.

The incidence of SCD increases with age, male gender and a history of cardiovascular disease. It also varies significantly geographically and according to the time frame. For a time frame of death occurring up to one hour after the onset of the first symptoms an overall yearly incidence of about 1 per 1000 of the total population is assumed reasonable for both Europe and the United States according to Priori et al. [21, p. 1377], making up about 13% of all natural deaths. Increasing the time frame to 24 hours the incidence increases to 18.5% following a report of the American College of Cardiology, the American Heart Association and the European Society of Cardiology [8, p. e395].

According to these figures, about 5000-6000 Norwegians die annually from SCD. Cause of death is registered for every individual in Norway and collected in the Norwegian Cause of Death Registry (no: Dødsårsaksregisteret) [9]. In 2012 this registry counted 41 913 deaths. Of these 13 010 people were registered to die from cardiovascular disease (no: “sykdommer i sirkulasjonsorganene”), which is about 31%, matching well up to the stated 30% of Mehra [20, p. S118].

## 2.2 The Heart

The heart is a muscle. A human heart is approximately the size of a fist and contracts about 60-80 times every minute on average. An overview of the heart anatomy can be seen in Figure 2.1. Most of the heart is made up of four cavities: the left and right atria at the top, and the left and right ventricles at the bottom. During a heart cycle these cavities fill with blood and then empty as they pump the blood out in the body.

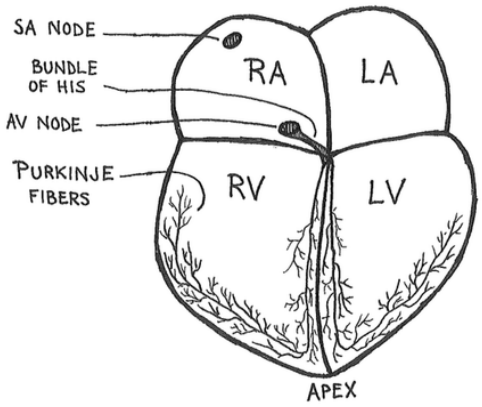



**Figure 2.1:** Schematic image of the heart showing the heart chambers and the conduction system. RA = right atrium, LA = left atrium, RV = right ventricle, LV = left ventricle.

### 2.2.1 The Circulatory System

The circulatory system is made up of the heart, the lungs and the blood vessels. It is separated into the pulmonary system, which brings blood between the heart and the lungs, and the systemic system, which brings blood between the heart and all parts of the body except the lungs. Together they provide the body with the oxygen it needs.

Oxygen depleted blood from the blood vessels fills the right atrium. From there it first flows passively into the right ventricle, before the atrium contracts, pumping some more blood actively into the right ventricle. When the right ventricle contracts, the blood is pumped into the lungs, where it is filled with oxygen. From the lungs the now oxygen enriched blood flows into the left atrium, from where it flows and is pumped into the left ventricle. When the left ventricle contracts, blood rich on oxygen is pumped through the blood vessels to reach out into the whole body. The atria contracts simultaneously, as do the ventricles.

### 2.2.2 The Conduction System

The contraction of the heart cavities is controlled by an electrical signal spreading through the conduction system of the heart. This signal starts in the sinoatrial node (SA node), located in the upper back of the right atrium. The SA node is also called the heart's pacemaker, due to it being self-oscillatory. From this node, the signal spreads through the atria. Reaching the atrioventricular node (AV node), at the border between the atria and the ventricles, the signal slows down a bit, causing the atria to contract before the ventricles. Then the signal spreads from the SA node through the bundle of His and the Purkinje network. The ends of the Purkinje network are connected to the endocardium of the ventricles, which is the inner surface layer of the cavities. From these connection sites the electrical signal spreads through the myocardium, which is the muscle tissue of the heart that makes up most of the heart wall, causing the heart to contract.

All the cells in the heart's conduction system are potential pacemaker cells, but the signal frequency is highest in the SA node and decreases moving away from it. This means that the signal from the SA node will trigger a new activation front in the rest of the conduction system before the other cells have time to self-activate, and the other cells only become pacemaker cells if the cells before them in the system fail.

The heart muscle consists of fibres, and the fibres are organized in sheets. The conductivity is higher in the direction of the fibres than perpendicular to them, and the conductivity in the sheets and perpendicular to them also varies. Thus the conductive properties of the heart muscle tissue are strongly anisotropic. Purkinje cells conduct electrical signals faster than other types of cells, about 4 m/s in the fiber direction as opposed to about 0.5 m/s in the fiber direction for ventricular myocytes (Klabunde [16]), thus the signal is spread fast to the whole ventricles, and all the muscle tissue contracts at almost the same time.

## 2.3 The Ventricles

During normal activation of the ventricles, the electrical signal moves from the bundle of His through the Purkinje network, and from there it spreads to the ventricular myocyte cells at the Purkinje-myocardial junctions (PMJs). These junctions are spread out in the ventricular muscle tissue to ensure fast activation of the whole tissue by stimulating many areas almost simultaneously. The activation of the tissue happens mainly from the apex to the base. The activation times vary between persons and also for the same person, for

example during different activities, but in this thesis, the activation time for the complete Purkinje system at rest is assumed to be about 25 ms, and the time from His bundle stimulation until the full activation of the ventricular mass is assumed to be about 70 ms (based on Behradfar et al. (rabbit), Ramanathan et al. (human) and Klabunde (human) [2, 22, 16], respectively).

When the signal moves through the Purkinje network to the ventricular muscle tissue in the normal fashion it is called anterograde propagation. The direction of the signal in each branch is then what is seen as the normal direction. Retrograde propagation is when the signal moves in the direction opposite to the normal direction. This happens when the signal goes from the myocardial tissue into the Purkinje network. In the Purkinje network this will give retrograde propagation in some branches and anterograde propagation in others. Retrograde propagation can arise during arrhythmia, and contribute to the arrhythmia being sustained and even fortified.

The ventricular muscle tissue consists of three main layers. From the inside and out these are the endocardium, the myocardium and the epicardium. The PMJs are sites found on the subendocardial surface, where the muscle tissue is activated from the Purkinje network. Between the Purkinje cells and the ventricular muscle cells there are cells of another type, called transitional cells. The transitional cells couple the ends of the Purkinje network to several ventricular muscle cells. There is a relatively high-resistance barrier between the Purkinje cells and the transitional cells and between the transitional cells and the ventricular muscle cells leading to a signal delay at the PMJs, see Tranum-Jensen et al. [25] and Rawling and Joyner [1]. Retrograde propagation is highly favored over anterograde propagation, as it is much easier to excite the one dimensional Purkinje fibres than the much greater ventricular mass. Behradfar et al. (rabbit) [2] considers anterograde delays in the range 4-14 ms and retrograde delays in the range 2-4 ms to be physiological, while Boyle et al. (rabbit) [3] operates with a lower retrograde delay (0.89 ms).

## 2.4 Cell Physiology

To be able to model the electrical activity in the cardiac tissue, it is necessary to be able to model the electrical activity in a single cell. There are two types of cells being modeled in the present problem: cardiac Purkinje cells and ventricular cardiomyocytes. These are both cardiac muscle cells, but they differ somewhat in cellular structure and electrical properties, so they require different cell models.

### 2.4.1 Anatomy

A muscle consists of many cylinder-shaped fascicles, which again are built from many muscle cells. A muscle cell, also called a muscle fiber, is shaped like a cylinder. Its most dominating entity is the myofibrils. These are cylinder-shaped structures running the length of the cell. Between the myofibrils and the other structures in the cell there is a liquid called cytoplasm. There are many myofibrils in one mucle cell and together with the cytoplasm they make up what is called the myoplasm. The myofibrils are built of many shorter cylinders called sarcomeres. These are the contractile units of the cell. The sarcomeres of the different myofibrils are approximately level. They are built of up thin and thick filaments, which are strands of different protein types. The thin filaments consist of actin, troponin and tropomyosin, while the thick filaments consist of myosin. Troponin can bind calcium ions, and is thus a buffer. The myoplasm also has a buffer type called calmodulin. An illustration of a muscle cell can be seen in Figure 2.2.

Around the myofibrils there is a tubular network called the sarcoplasmic reticulum (SR). The SR contains enlarged segments called cisternae, where calcium ions bind to the protein buffer calsequestrin. When triggered, the SR can release calcium from these cisternae into the myoplasm. The calcium is taken up again in the SR by the rest of the SR network, before it diffuses back into the cisternae. The cisternae are located at specific places along the sarcomeres.

The cell is surrounded by a cell membrane called the sarcolemma. In some cell types the sarcolemma dimples into the cell in some places, creating membrane tubules around the myofibrils that can stretch quite long into the cell. These tubules are called transverse tubules or T-tubules, and in those cardiac cells that have them they are normally opposed to a cisternae in the SR that may wrap around it, creating a structure called a diad. The T-tubules enable extracellular ions to rapidly reach to the center of the cell. Ventricular myocytes have T-tubules, while Purkinje cells do not.

In addition to the mentioned structures, the cardiac cells also contain one nucleus (while skeletal muscle cells contain several nuclei), and many mitochondria, placed among the myofibrils.

### 2.4.2 The Cell Membrane

The cell membrane separates the intracellular space from the extracellular space. The membrane itself is impermeable to ions, but it has embedded in it some large proteins acting as specialized transport channels, the different proteins only letting certain ions pass. These ion channels can open and close

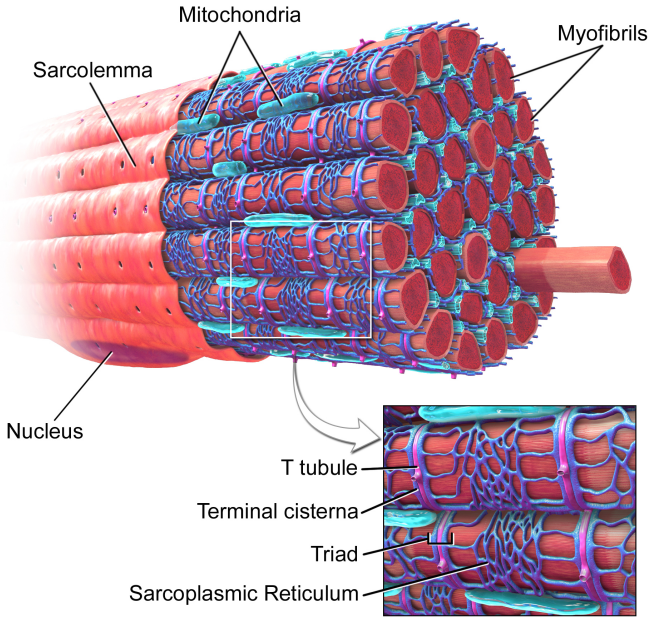



**Figure 2.2:** Schematic image of a muscle cell[1]. The illustration is of a skeletal muscle cell. In cardiac muscle cells the structure named "triad" in the figure is named "diad", and consists of a T-tubule and one single cisternae, as opposed to a T-tubule and two cisternae in skeletal muscle cells. In addition cardiac muscle cells only has one nucleus, while skeletal muscle cells have several nuclei.

in response to changes in the electric field or the ionic concentrations. They can consist of several subunits and can portray a quite complex behaviour.

The passage of ions through an ion channel might be passive, the flow being driven by concentration gradients or electric fields, or it may be active, when the ions are being transported against mentioned gradients or fields. Active transport proteins are called pumps if they transport only one type of ion, using chemically stored energy in the cell to do so. If they "exchange" two types of ions, using the concentration gradient of one of the types to pump the other type against its concentration gradient they are called exchangers.

Since the ion channels are specialized for certain ion types, and can open and close as a reaction to changes in its environment, it is possible for a potential difference to build up across the cell membrane. Let us say we start out with two types of ions, oppositely charged, both present on the inside and outside of the cell. We assume both to have a higher concentration on the outside. Then the concentration gradients of both will work to bring the

[1] http://commons.wikimedia.org/wiki/File:Blausen_0801_SkeletalMuscle.png

ions inside the cell. If the cell membrane only permits the entrance of one of the ion types, this will create a potential difference across the membrane, working to bring the ions entering the cell back outside. Thus the diffusive and electrical powers work in opposite directions, and an equilibrium will be reached where ions no longer flow. In this resting state, the cardiac cell will have different internal and external ionic concentrations, resulting in a potential difference across the cell membrane, called the transmembrane potential, or simply membrane potential.

In this thesis the transmembrane potential is denoted by $V$. An $i$ or $e$ as subscript signifies that something is intracellular or extracellular, respectively. The transmembrane potential is defined as the intracellular potential minus the extracellular potential: $V = V_i - V_e$. The value of the transmembrane potential that gives zero flux is called the Nernst equilibrium potential, or the reversal potential. This will be further described in Chapter 3.

### 2.4.3 Action Potential and Ionic Currents

Cardiac cells are excitable. If the transmembrane potential is brought above a threshold value this will trigger what is called an action potential. Instead of just returning to resting state, the membrane potential will get more positive very fast, until it is close to, or even well above, zero. This is called depolarization. From here the membrane potential will slowly start to fall, making its way back to the resting state. This is called repolarization. Sometimes the potential will reach a slightly lower value than the resting state before returning to rest. This is called hyperpolarization. The complete potential excursion from the onset of the depolarization until the potential is back to its resting value is called an action potential, and is illustrated in Figure 2.3 It can be split into five phases, in which different ions and currents dominate. The initial depolarization can come from an external stimulus, or for pacemaker cells it can be caused by transmembrane currents controlled by the cell itself.

- Phase 0: Depolarization.

  Phase 0 starts by a depolarization bringing the transmembrane potential above a threshold value. This allows the fast $Na^+$ channels to open, causing a rapid influx of $Na^+$ ions by the fast $Na^+$ current $I_{Na}$, which raises the membrane potential significantly (from about -85 mV to about 40 mV for the two cell types of interest here) in a very short time (about 1 ms).

- Phase 1: First part of the repolarization.

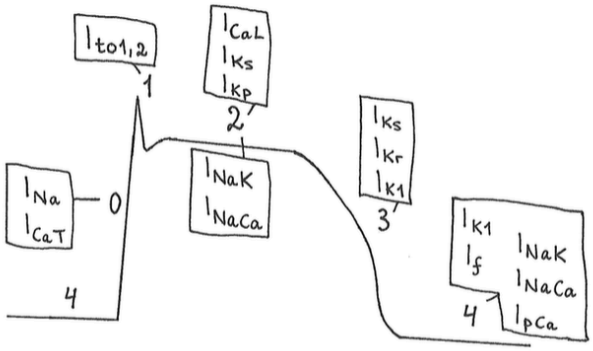



**Figure 2.3:** Schematic illustration of an action potential. Phases and main currents are shown. Time is on the first axis, typically a range of a few hundred ms. Voltage is on the second axis, typically in the range from about -85 mV to about 40 mV.

When the transmembrane potential approaches the $Na^+$ equilibrium potential, the fast $Na^+$ channels are inactivated by a slow inactivation gate. This is modeled by the slowly inactivating late $Na^+$ current $I_{NaL}$. At the same time, the 4AP-sensitive transient outward $K^+$ current $I_{to1}$ and the $Ca^{2+}$ dependent transient outward $Cl^-$ current $I_{to2}$ (the latter only in the ventricular myocyte model) start carrying $K^+$ and $Cl^-$ ions, respectively, out of the cell. For a little while the net current is outwards, bringing the transmembrane potential down a little. The $Na^+$ channels stay in the inactivated state well into phase 3. This makes it impossible to initiate a new action potential from the beginning of phase 0 until the $Na^+$ channels leave this state. This period is called the refractory period.

- Phase 2: "Plateau" phase, second part of the repolarization.

  Shortly after the inactivation of the fast $Na^+$ channels, the $Ca^{2+}$ current through the L-type $Ca^{2+}$ channel, $I_{CaL}$, starts bringing $Ca^{2+}$ ions into the cell, working against the efflux of positive $K^+$ ions, which are now carried by the slow delayed rectifier $K^+$ current $I_{Ks}$, giving the action potential a "plateau" shape. In the ventricular myocytes the plateau $K^+$ current $I_{Kp}$ contributes to the plateau shape.

- Phase 3: "Rapid repolarization" phase, last repolarization part.

  The L-type $Ca^{2+}$ channels depend on the membrane potential and local $Ca^{2+}$ concentrations in the cell. After a while they will close, while $I_{Ks}$ continues to bring $K^+$ ions out of the cell, in addition to new $K^+$

channels opening. Rapid $K^+$ efflux happens through the rapid delayed rectifier $K^+$ current $I_{Kr}$, while the time-independent $K^+$ current $I_{K1}$ gives $K^+$ influx against the concentration gradient, but the net current is still outward, and the repolarization happens fast.

- Phase 4: Diastolic depolarization phase. Resting phase.

  When the cell is fully repolarized, the transmembrane potential is back to its resting state. Pacemaker cells such as the Purkinje cell then slowly starts to depolarize because of the hyperpolarization activated $Na^+$-$K^+$ current $I_f$, while non-pacemaker cells such as the ventricular myocyte stay relaxed until receiving a new external stimulus, and a new action potential is triggered.

The $Na^+$-$K^+$ pump $I_{NaK}$ and the $Na^+$-$Ca^{2+}$ exchanger $I_{NaCa}$ are active in phases 2 and 4. Their job is to maintain the membrane potential while bringing the ionic concentrations back to the values they had before depolarization. During an action potential, $Na^+$ ions enter the cell while $K^+$ ions leave, so $I_{NaK}$ gives an efflux of $Na^+$ ions and an influx of $K^+$ ions to restore balance. The exchange rate is three $Na^+$ ions to two $K^+$ ions. Both ion types are carried against their concentration gradient. The $Na^+$-$Ca^{2+}$ exchanger is one of the most important mechanisms for removing $Ca^{2+}$ ions from the intracellular space. It brings one $Ca^{2+}$ ion out of the cell and three $Na^+$ ions into it. To bring the $Na^+$ concentration back to its original value, $I_{NaK}$ must be greater than $I_{NaCa}$, since $I_{NaCa}$ brings $Na^+$ ions back into the cell. In addition to this, the $I_{K1}$ current works along $I_{NaK}$ in giving an influx of $K^+$ ions, being active through phases 3 and 4, and the sarcolemmal $Ca^{2+}$ pump current $I_{pCa}$ works to remove $Ca^{2+}$ ions from the cell alongside $I_{NaCa}$.

There are two types of currents included in the Purkinje cell model but not in the ventricular myocyte model, these are the $Ca^{2+}$ current through the T-type $Ca^{2+}$ channel $I_{CaT}$ and $I_f$. In phase 2, the L-type $Ca^{2+}$ current (L for long-lasting) is important to give the action potential a plateau. Pacemaker cells also have a T-type $Ca^{2+}$current (T for transient) $I_{CaT}$. The T-type channels react to lower values of the membrane potential than the L-type and open faster but close earlier. The consequence is that it plays a part in initiating an action potential, rather than prolonging it. Since pacemaker cells are able to spontaneously depolarize, it makes sense that they have more mechanisms to get the depolarization started. Purkinje cells are latent pacemaker cells, so they have T-type $Ca^{2+}$ channels, while ventricular myocytes do not.

The current responsible for the depolarization of pacemaker cells is the $I_f$, which is a mixed $Na^+$-$K^+$ current (in the model the contributions from

the different types of ions are separated into two separate currents). The channels of this current conduct more current as the membrane potential becomes more negative. This current is also called the funny current or the pacemaker current.

The cell models for the ventricular myocyte and the Purkinje cell both assume that there are always some $Ca^{2+}$ and $Na^{+}$ ions passing the membrane. The resulting currents are called background currents, or leakage currents, $I_{Cab}$ and $I_{Nab}$ for the $Ca^{2+}$ and $Na^{+}$ background currents, respectively. The model for a ventricular myocyte also models a background $Cl^{-}$ current $I_{Clb}$. In the ventricular myocytes, cotransporter fluxes $CT_{NaCl}$ and $CT_{KCl}$ are required alongside $I_{Nab}$ to maintain the correct $Na^{+}$ concentration during the resting state.

### 2.4.4 Excitation-Contraction Coupling

The sarcomeres are the contractile units of the muscle cells. They all contract simultaneously and when they do, the whole cell becomes shorter. The sarcomeres are built of thin and thick filaments, both types of constant length. In resting state, the thin and thick filaments only overlap slightly - the thick filaments in the center of the sarcomere, the thin filaments further out - but during contraction they slide along the length of the myofibrils until they overlap completely, shortening the length of the sarcomere. This sliding happens as part of the cross bridge cycle.

As stated earlier, the thin filaments consist of the proteins actin, troponin and tropomyosin, where troponin can bind calcium, while the thick filaments consist of the protein myosin. Actin in the thin filaments has binding sites for the myosin heads of the thick filaments. When the cell is in its resting state these binding sites are blocked by tropomyosin, but if calcium binds to troponin, troponin changes form and moves tropomyosin, letting the myosin heads bind to actin in so-called cross bridges. Then the cross bridges move, sliding actin towards the center of the sarcomere, before the myosin head lets go of actin. This is repeated as long as the myosin binding sites on actin are not covered by tropomyosin, i.e. as long as calcium is present, making the cell shorter and shorter.

Calcium enters the cell during phase 2 of the action potential, the plateau phase. This calcium is not enough to start the cross bridge cycle, but it triggers the release of much more calcium from the cisternae in the SR. This is enough for the cycle to begin and it does not stop before well into phase 4, the resting phase. The same amount of calcium as entered the intracellular space from the outside is then removed, and the rest is taken up again by the SR. The calcium released from the SR influences the L-type calcium current.

### 2.4.5 Cell Connectivity

Muscle cells are connected by large proteins embedded in the cell membrane. These proteins form channels between adjacent cells, forming direct contact between the inner spaces of the cells. These connections are called gap junctions. Via the gap junctions, the ions leading to the depolarization of one cell may move on to the next cell bringing about its depolarization as well. In this way, an electrical stimulus in a small area of the heart may lead to a depolarizing wave through the whole heart.

## 2.5 What to Model

The contraction of the heart, which controls the blood supply to the whole body, is controlled by electrical signals in the heart's conduction system and muscle tissue. This electrical signal is a difference in voltage across the cell membrane, which can come into being because the cell membrane is selectively permeable to different ion types. Therefore, to model the electrical activity, one must model the transmembrane potential of the cells. In a single cell the transmembrane potential is controlled by transmembrane ionic currents, and these again may depend on concentrations of different ion types in different parts of the intracellular area, so these must also be modeled, as well as intracellular fluxes. For a specific cell type, all these quantities are described by a cell model. When looking at a tissue, diffusion between the cells must be considered in addition. This is modeled by a tissue model, which must be coupled together with a cell model.

# Chapter 3

# Mathematical Models

In the previous chapter the biology of the problem was described, and it was defined more accurately what it is desirable to model. The next step is to find a way to model this in a way that gives us the necessary information. A mathematical framework is perfect for this. In essence, everything that is studied in this problem is how quantities change in space or time. If the cause of change of a quantity is known, differential equations can be used to model its development. If all causes of change are known and represented correctly, and there are also sufficient initial conditions to define the system properly, differential equations hold the complete information about the quantity at all times and/or in all space, depending on what the problem is. So, given the equations and appropriate initial conditions in the form of numbers, numbers representing the quantity are given in return. As stated in Section 2.5, the goal is to model the processes in a single cell, as well as those on tissue level. Both of these types of models will be considered: the PRd model [17] for a Purkinje cell, the Decker et al. model [15] for a ventricular myocyte, and the monodomain equation for cardiac tissue. In addition two simpler cell models will be introduced: a cubic polynomial and the FitzHugh-Nagumo (FHN) model.

## 3.1 Single Cell Models

The goal is to model the transmembrane potential $V$ and how it changes in time. To find a general expression for this, the procedure in Lines's doctoral thesis [18, Section 3.1] was followed.

The cell membrane has properties corresponding to both a resistor and a capacitor, see Figure 3.1 for an electric circuit model of the cell membrane. Motivated by this, the transmembrane current $I_m$ [$\mu$A/cm$^2$] is modeled as

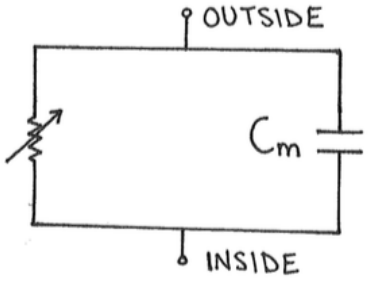



**Figure 3.1:** Electrical circuit model of the cell membrane consisting of a resistance and a capacitance.

the sum of a capacitive current $I_c$ and a resistive current $I_{ion}$:

$$I_m = I_c + I_{ion}. \tag{3.1}$$

The resistive current is called $I_{ion}$ because it carries ions across the membrane, while the capacitive current represents the accumulation of charge across the membrane, and doesn't move any ions.

The electrical potential across the membrane can be written as:

$$\Delta V = \frac{\Delta q}{C_m}, \tag{3.2}$$

where $q$ is the charge [nC/cm$^2$] and $C_m$ is the capacitance [$\mu$F/cm$^2$] of the membrane.

Current is defined as:

$$I = \lim_{\Delta t \to 0} \frac{\Delta q}{\Delta t}, \tag{3.3}$$

where $t$ is time [ms].

Dividing equation (3.2) by $\Delta t$ and letting $\Delta t \to 0$ gives a relation between the change of the transmembrane potential and the capacitive current:

$$\frac{dV}{dt} = \frac{1}{C_m} I_c = -\frac{1}{C_m}(I_{ion} - I_m). \tag{3.4}$$

Often, $C_m$ is incorporated into the ionic currents, so the ionic currents $I$ are measured in $\mu$A/$\mu$F instead of $\mu$A/cm$^2$. This will be the case for the currents in the rest of this thesis. In accordance with the naming in the model, $I_{ion}$ is renamed $I_{tot}$ and $I_m$ is renamed $I_{app}$, as this current corresponds to external stimulus. The final equation is then:

$$\frac{dV}{dt} = -(I_{tot}(t, s) - I_{app}(t)), \tag{3.5}$$

where $s$ denotes the state vector. This equation is the basis for all the cell models.

### 3.1.1 A Cubic Polynomial

The simplest cardiac cell model used in this thesis use a direct expression for the right hand side of equation (3.5), so this model only consists of one equation:

$$\frac{dV}{dt} = f(V), \tag{3.6}$$

where $f$ is:

$$f(V) = -A^2(V - V_{rest})(V - V_{th})(V - V_{peak}). \tag{3.7}$$

The cell model with this cubic polynomial as the right hand side of equation (3.5) is the simplest cell model that reproduces the main behaviour of the depolarization phase of the action potential.

The function $f$ has three zeros: $V_{rest}$, $V_{th}$ and $V_{peak}$. These values are equilibrium points of $V$ in equation (3.6), since $f = 0$ does not contribute any change to $V$. If the initial value of $V$ is smaller than $V_{th}$, $V$ will tend towards $V_{rest}$. Physiologically this corresponds to the cell receiving a small stimulus. When the stimulus is removed, the cell will return to the resting potential, here called $V_{rest}$. If $V$ starts out greater than $V_{th}$, $V$ will move towards $V_{peak}$. This corresponds to the cell receiving a stimulus large enough to bring the transmembrane potential across the threshold value, here named $V_{th}$. Then the cell enters its depolarization phase, the transmembrane potential growing until it reaches a peak, here labeled $V_{peak}$.

Having reached its peak, the transmembrane potential will enter its repolarization phase, where it returns to the resting state. The cubic $f$ given here is not able to reproduce this phase, since once the peak potential $V_{peak}$ is reached, the transmembrane potential does not change any more.

The parameter $A$ [$\mathrm{mV}^{-1}$ $\mathrm{ms}^{-1/2}$] controls the magnitude of $f$, and thus the upstroke velocity of $V$. A high value of $A$ gives a high change rate and therefore a steep depolarization front. Physiologically $A$ represents the permeability of the membrane. A high value means a high permeability, meaning that more ions are able to cross the membrane than with a low permeability, leading to faster changes.

### 3.1.2 The FHN Model

The next cell model to be considered is the FitzHugh-Nagumo (FHN) cell model. It consists of two equations with two unknown variables. The equations are defined as by Keener and Sneyd [14, eqs.(5.38)-(5.40)]. The first equation corresponds to the basis equation (3.5):

$$\epsilon \frac{dV}{dt} = f(V) - w + I_{app}, \tag{3.8}$$

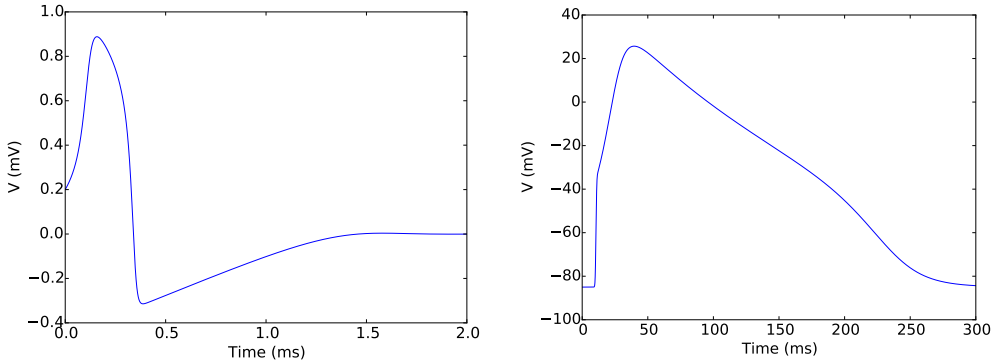
$$\frac{dw}{dt} = V - \gamma w, \tag{3.9}$$

where

$$f(V) = -V(V - \alpha)(V - 1) \tag{3.10}$$

for $0 < \alpha < 1$, $\epsilon \ll 1$, where $\gamma$ and $\epsilon$ are scalars and $\alpha$ a potential [mV].

The function $f$ is on the same form as the $f$ in the previous section, only with $A = 1$ mV$^{-1}$ ms$^{1/2}$), $V_{rest} = 0$ mV, $V_{th} = \alpha$ mV and $V_{peak} = 1$ mV. While the cubic polynomial alone was only able to reproduce the characteristics of the depolarization phase of the action potential, the FHN model is extended to also incorporate the repolarization phase, as can be seen in Figure 3.2a. This is done by adding a second variable $w$ [$\mu$A/$\mu$F], called the recovery variable since it enables the recovery of the membrane potential to its resting potential. The idea is that $w$ grows along with $V$, and at the same time $w$ is subtracted from the rate of change of $V$, eventually making $V$ decrease. Depolarization followed by repolarization now gives a wave front.




**(a)** Original. **(b)** Modified.

**Figure 3.2:** Action potential for the FHN cell model. (a) Original model with $V(0) = 0.2$ mV, $w(0) = 0.0$ $\mu$A/$\mu$F, $\gamma = 0.5$, $\epsilon = 0.01$, $\alpha = 0.1$ mV. (b) Modified model with parameters defined in the source code [12].

The varible $I_{app}$ signifies an external stimulus applied to the cell. To achieve depolarization the transmembrane potential must either have an initial value higher than the threshold value, or there must be an external stimulus.

A small value for the scalar $\epsilon$ makes $V$ a fast variable compared to $w$, since the change in transmembrane potential is magnified. This compensates somewhat for the FHN model having a slower upstroke than a real cell.

A non-physiological feature of the FHN model is that the transmembrane potential hyperpolarizes, i.e., at the end of the repolarization phase the potential falls lower than the resting potential before returning to it. As a remedy for this, Rogers and McCulloch, as referred by Sundnes et al. [24, p.35], suggested that the term with the recovery variable $w$ in equation (3.8) be multiplied with the membrane potential variable $V$. The new equation becomes:

$$\epsilon \frac{dV}{dt} = f(V) - Vw + I_{app}. \tag{3.11}$$

Since the modified term will now approach zero as $V$ approaches the resting potential (here zero), $V$ will no longer reach sub-zero values.

In both the original and the modified FHN equations, the transmembrane potential has values between 0 and 1 mV. The equations can be reparameterized to take on more physiological values by defining:

$$V^* = V_{amp}V + V_{rest}, \tag{3.12}$$

where $V_{amp} = V_{peak} - V_{rest}$ (using only the values of $V_{peak}$ and $V_{rest}$, so $V_{amp}$ is dimensionless). Solving this equation for $V$ and inserting into either the original or modified equations will give equations with values for $V$ (now called $V^*$) in the desired range. The recovery variable $w$ has no physiological interpretation and does not need scaling, but for notational convenience it can be useful to replace it by the new variable $W$ defined as:

$$W = V_{amp}w. \tag{3.13}$$

An action potential for the modified FHN cell model with physiological values can be seen in Figure 3.2b.

### 3.1.3 The PRd Model

The PRd cell model is a lot more complicated than the two previous models. While the cubic polynomial and the FHN equations are constructed to reproduce observed behavior of cardiac cells on a macroscopic level, the PRd model tries to describe the physiology of the cells. Only in this way is it possible to investigate how changes in the physiology of the cells affect the heart tissue, and possibly the whole heart. This section will present an overview of the model and try to explain some of the physiology behind the equations, but to know the complete model, the reader is referred to the original publication by Li and Rudy [17].

The model describes a cardiac Purkinje cell. What is new about this model is that the the authors have developed a mathematical model of the Purkinje cell that represents its unique electrophysiological and calcium handling properties instead of adopting that of ventricular myocytes. The complete model consists of 38 ordinary differential equations (ODEs), many expressions and 80 parameters. The state vector consists of the transmembrane potential, 11 ionic concentrations, 23 gating variables, two fluxes and one variable that is involved in the expression for a flux but is not a flux itself. The state variables can be seen in Table 3.1.

| | | | |
|---|---|---|---|
| $V$ | $[K^+]_i$ | $xs1$ | $j_L$ |
| $[Ca^{2+}]_{PCS}$ | $[CAMK]_{trap}$ | $xs2$ | $j_{L3}$ |
| $[Ca^{2+}]_{SSL}$ | $m$ | $xr$ | $b$ |
| $[Ca^{2+}]_{JSR}$ | $h$ | $a$ | $g$ |
| $[Ca^{2+}]_{CSR}$ | $j$ | $i$ | $y$ |
| $[Ca^{2+}]_{NSR}$ | $d$ | $i2$ | $u_{IP_3R}$ |
| $[Ca^{2+}]_i$ | $f$ | $m_L$ | $J_{RyR_2}$ |
| $[Na^+]_{PCS}$ | $f2$ | $m_{L3}$ | $J_{RyR_3}$ |
| $[Na^+]_{SSL}$ | $fca$ | $h_L$ | |
| $[Na^+]_i$ | $fca2$ | $h_{L3}$ | |

**Table 3.1:** State variables in the PRd model. $V$ is the transmembrane potential, $[X]_y$ is the concentration of X in compartment y, $J_{RyR_2}$ and $J_{RyR_2}$ are fluxes, $u_{IP_3R}$ is involved in the expression for the $J_{IP_3}$ flux, and the rest are gating variables involved in the current expressions.

The equation for the transmembrane potential is on the form of equation (3.5), which is restated here for convenience:

$$\frac{dV}{dt} = -(I_{tot}(t,s) - I_{app}(t)). \tag{3.14}$$

An action potential for a Purkinje cell modeled by the PRd cell model can be seen in Figure 3.3.

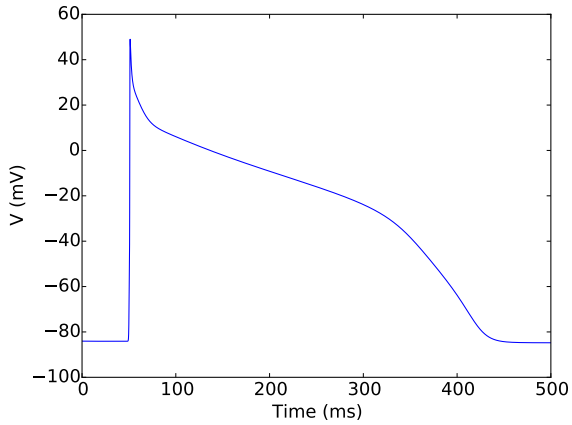



**Figure 3.3:** An action potential of a Purkinje cell for the PRd model.

### Cell Structure

The PRd model is a compartmental model. This means that the cell volume is divided into separate parts and the ionic concentrations within each compartment are assumed to be constant. Changes in the concentrations happen as a result of fluxes between the compartments and ionic currents across the cell membrane. Areas that are not adjacent can be modeled as a single compartment, given that they have the same ionic concentrations and these change similarly in time. The ionic flows in the model should have realistic physiological values.

The PRd cell model divides the cell into six compartments: the peripheral coupling subspace (PCS), the subsarcolemmal compartment (SSL), the bulk myoplasm (Myo), the junctional sarcoplasmic reticulum (JSR), the network sarcoplasmic reticulum (NSR) and the corbular sarcoplasmic reticulum (CSR). An overview of the cell model is shown in Figure 3.4.

The volume immediately below the cell membrane is divided into two compartments: PCS is a small compartment where the L-type $Ca^{2+}$ channels and a small portion of the sodium-calcium-exchangers are placed, while SSL is the rest of this area, where the rest of the ion channels are placed. The myoplasma has its own compartment Myo, while the SR is divided into three compartments: the NSR takes up calcium from the Myo and the SSL compartments, from there it diffuses into the JSR and the CSR, which release calcium into the Myo and the PCS, respectively, when triggered.

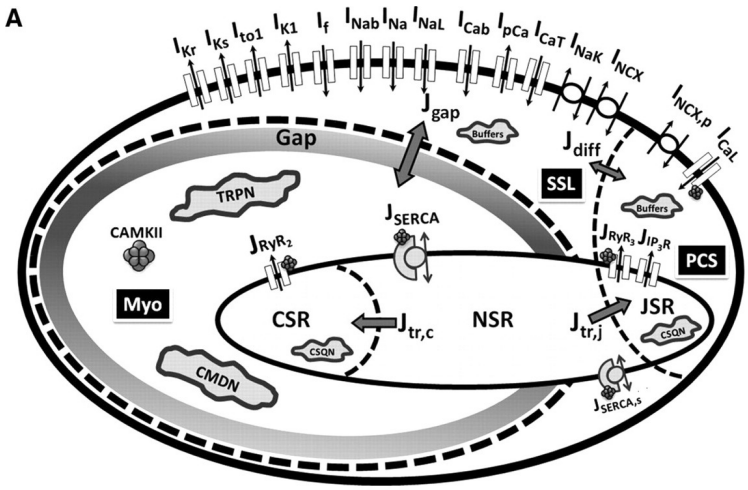



**Figure 3.4:** A graphic overview of the PRd cell model for a canine cardiac Purkinje cell. This is Figure 1.A from page 73 in the article "A Model of Canine Purkinje Cell Electrophysiology and $Ca^{2+}$ Cycling. Rate Dependence, Triggered Activity, and Comparison to Ventricular Myocytes", by authors Pan Li and Yoram Rudy, published in *Circulation Research* 2011;109;71-79 [17]. Reproduced with permission.

There are different types of calcium buffers in all the compartments except the NSR: calsequestrin (CSQN) in the JSR and the CSR, troponin (TRPN) and calmodulin (CMDN) in the Myo and (in the figure) unspecified buffer proteins in the PCS and the SSL. The different transmembrane currents and intracellular fluxes can be seen in the overview of the cell model in Figure 3.4.

### Channel Gating

In Section 2.4.2 it was stated that the ion channels in the cell membrane can open or close in response to the electrical field or ionic concentrations. The mathematical models used in this thesis to describe the ion channels consider the channels to consist of several sub-units, each of which can be open or closed. The channel is only open for ions to pass when all units are open.

Looking at a single unit, the change between the open and closed state may be expressed as:

$$C \underset{\beta}{\overset{\alpha}{\rightleftarrows}} O, \tag{3.15}$$

where $O$ and $C$ are the open and closed states, respectively, and $\alpha$ and $\beta$ [1/ms] are the rates of opening and closing, respectively. The rate constants often depend on the membrane potential. The law of mass action (see for example Keener and Sneyd [14]) states that in a reaction of this kind, the rate of change from one state to the other is proportional to the concentration of channel units in the state of origin. With $[O]$ and $[C]$ denoting the concentration of open and closed states, respectively, this results in:

$$\frac{d[O]}{dt} = \alpha(V)[C] - \beta(V)[O]. \tag{3.16}$$

Dividing this by the total concentration of channel units $[O] + [C]$, which is assumed to be constant, gives an equation for the portion of open channel units $g = [O]/([O] + [C])$:

$$\frac{dg}{dt} = \alpha_g(V)(1 - g) - \beta_g(V)g, \tag{3.17}$$

or equivalently:

$$\frac{dg}{dt} = \frac{g_\infty - g}{g_\tau}, \tag{3.18}$$

where $g_\infty = \alpha/(\alpha + \beta)$ and $g_\tau = 1/(\alpha + \beta)$. The variable $g$ can also be interpreted as the probability that the unit is open, and it is called a gating variable.

Since $\alpha$ and $\beta$ depend on $V$ it is impossible to find a general solution of equation (3.18), but with $\alpha$ and $\beta$ constant, the analytical solution is:

$$g(t) = g_\infty + (g_0 - g_\infty)e^{-t/g_\tau}, \tag{3.19}$$

where $g_0$ is the initial value of $g$. As $t$ increases, $g$ will approach $g_\infty$, the rate of change depending on $g_\tau$. If $g_\tau$ is small the change will be fast and $g_\infty$ can be a reasonable approximation to $g$. This approximation is used in the equations for the currents $I_{Kr}$, $I_{K1}$ and parts of $I_{to1}$.

The channel gates can be slow or fast activation or inactivation gates. Assuming that the channel units open and close independent of each other, the probability $O$ that the channel is open is the product of the probabilities that all the channel units are open. These probabilities can all be written on the form of equations (3.17) or (3.18), but the rate constants $\alpha$ and $\beta$ may vary from one channel unit to the next. Since $g$ and $O$ are probabilities they must have values in the interval $[0, 1]$.

The expressions for $\alpha$ and $\beta$ have no physiological meaning. They are constructed to match experimental data and cannot be expressed in any common general form. Given the expressions for the rate constants, and

initial conditions, the ODEs for the gating variables are properly defined. If $g_\infty$ is used directly to approximate $g$ no ODE is required. For the explicit expressions for the rate constants and initial conditions in the PRd model, the reader is referred to the original publication [17].

### Ionic Currents

Three types of ions are included in the model: sodium, potassium and calcium. The total transmembrane current is described as a sum of the currents of these individual ions, so equation (3.5) becomes:

$$\frac{dV}{dt} = -(I_{Na,tot} + I_{K,tot} + I_{Ca,tot}). \tag{3.20}$$

Each of these currents are again described as sums of different types of currents of the same ion. These ion flows can be passive flow through different types of membrane channels, or active flow in the form of pumps or exchangers, as described in Section 2.4.2. The equations for the total currents of the different ion types are:

$$I_{Na,tot} = I_{Na} + I_{NaL} + I_{Nab} + 3I_{NaK} + I_{f,Na} + 3(I_{NCX,SSL} + I_{NCX,PCS}), \tag{3.21}$$

$$I_{K,tot} = I_{Kr} + I_{Ks} + I_{K1} - 2I_{NaK} + I_{to1} + I_{f,k}, \tag{3.22}$$

$$I_{Ca,tot} = I_{CaL} + I_{Cab} + I_{pCa} + I_{CaT} - 2(I_{NCX,SSL} + I_{NCX,PCS}). \tag{3.23}$$

All the individual currents and their roles during an action potential are described in Section 2.4.3. In the overview of the cell model in Figure 3.4 the currents are depicted with directions. The intracellular potassium concentration is higher than the extracellular concentration, so this concentration gradient leads to efflux of potassium, while for sodium and calcium it is the other way around. During the action potential, the membrane potential will reach the equilibrium potentials of the different ions, and the directions of the ionic flows will be reversed. This holds for all the passive ion channels, where the flows are driven by concentration gradients and electrical effects.

The $Na^+$-$K^+$ pump makes three $Na^+$ ions leave the cell and two $K^+$ ions enter the cell, giving these two terms opposite signs. The same goes for the $Na^+$-$Ca^{2+}$ exchanger, that sends one $Ca^{2+}$ ion (with two positive charges) out of the cell and three $Na^+$ ions into the cell.

The concept of an equilibrium potential was introduced in Section 2.4.2. The Nernst equilibrium potential of a specific ion is the transmembrane potential that gives zero flux of that same ion across the membrane. A further increase in the transmembrane potential after this point is reached will give

an ion flux in the opposite direction of before, and for this reason it is also called the reversal potential. The Nernst equation can be expressed as by Sundnes et al. [24, eq.(2.52)]:

$$V_{eq} = \frac{RT}{zF}\, ln\left(\frac{c_e}{c_i}\right), \tag{3.24}$$

where $V_{eq}$ is the equilibrium potential, $R$ [J/(kmol K)] is the ideal gas constant, $T$ [K] is the temperature, $F$ [C/mol] is Faraday's constant, $z$ is the valence of the ion [dimensionless], and $c_e$ and $c_i$ [mmol/L] are the extra- and intracellular concentrations of the ion, respectively. The valence of an ion is its excess or deficiency of ions (with sign) compared to the neutral state and says something about the ion's ability to bind to other ions.

The currents through the passive channels can be modeled in several ways. Here two alternative models are used:

$$I = OG_{max}(V - V_{eq}), \tag{3.25}$$

$$I = \frac{1}{C_m} OP \frac{z^2F^2}{RT} V \frac{c_i - c_e\, exp(\frac{-zFV}{RT})}{1 - exp(\frac{-zFV}{RT})}, \tag{3.26}$$

where $O$ is the probability that all the subunits of the channel are open to let the flux through, as defined under the headline "Channel Gating" earlier in this section. For the background currents $O = 1$ since these currents are assumed to be present all the time. The variable $G_{max}$ [mS/$\mu$F] is the maximum conductance per area with $1/C_m$ incorporated (when all channels are open) and $P$ [cm/s] is the permeability of the cell membrane to the ion in question. The division by $C_m$ was explained in the beginning of this chapter.

The Nernst equation (3.24) holds in general, and one can see that $V = V_{eq}$ in both expressions for current above gives $I = 0$. The linear model given by equation (3.25) is the simplest formulation satisfying the Nernst equilibrium equation, and is constructed for this purpose. The more complex model given by equation (3.26) is obtained through a more physiologically meaningful process, which will not be further explained here (see for example Keener and Sneyd [14]). It is called the Goldman-Hodgkin-Katz (GHK) equation. All the passive currents in equations (3.21)-(3.23) are on the form of equations (3.25) or (3.26).

### Pumps and Exchangers

There are four types of pumps or exchangers in the PRd model: the SERCA pump, the $Na^+$-$Ca^{2+}$ exchanger, the $Na^+$-$K^+$ pump and the sarcolemmal calcium pump ($I_{pCa}$). The SERCA pump is treated in the next segment.

These pumps and exchangers can be described in a number of ways. Different models can have a different number of states, according to their detail level. Since the equations describing these currents are not as general as many of the other equations in the model, they are not further described. Details can be found in the original publication [17].

### SR $Ca^{2+}$ Fluxes

The PRd model has three fluxes modeling the release of $Ca^{2+}$ from the SR: $J_{RyR_3}$, $J_{RyR_2}$ and $J_{IP_3R}$. The formulations of these are modified from earlier models. For details about which models they are based on, the reader is referred to the original publication [17]. The fluxes in the cell model are defined with respect to a reference volume and have units mmol/(L ms).

The flux $J_{RyR_3}$ goes from JSR to PCS. It depends on $Ca^{2+}$ fluxes in the PCS compartment as well as $Ca^{2+}$ concentrations in the PCS and the JSR. The flux $J_{RyR_2}$ goes from CSR to Myo and depends on $Ca^{2+}$ fluxes in the Myo compartment and $Ca^{2+}$ concentrations in the Myo and CSR compartments. The formulation of the fluxes can be written in the following way, where $x$ is 2 or 3:

$$\frac{dJ_{RyR_x}}{dt} = \frac{RyR_{x_\infty} - J_{RyR_x}}{\tau_{RyR_x}}. \tag{3.27}$$

Here $RyR_{x_\infty}$ [mmol/(L ms)] depends on the change of the $Ca^{2+}$ concentration in the compartment that the respective flux releases $Ca^{2+}$ into. If the change is positive, $RyR_{x_\infty}$ is proportional to it, so an increase in the $Ca^{2+}$ concentration in these compartments trigger the release of more $Ca^{2+}$ from the SR. This is called $Ca^{2+}$ induced $Ca^{2+}$ release and is an important part of the cell dynamics during an action potential. The given equation has the same form as equation (3.18) for a gating variable, but since $RyR_{x_\infty}$ and $\tau_{RyR_x}$ [ms] cannot be written as functions of any $\alpha$ and $\beta$ in the specified way, these equations are different, and $J_{RyR_2}$ and $J_{RyR_3}$ are not gating variables.

The flux $J_{IP_3R}$ goes from JSR to PCS. The equations for this flux are on a different form than the equation for the $J_{RyR}$s. The ODE is not for $J_{IP_3R}$ directly, but for the variable $u_{IP_3R}$ included in the expression for $J_{IP_3R}$. This state variable depends on the $Ca^{2+}$ concentration in the PCS compartment, which is where the $J_{IP_3R}$ flux releases $Ca^{2+}$, but the expression for $J_{IP_3R}$ is more complex than those for the $J_{RyR}$ fluxes, and will not be further discussed.

The SERCA pumps bring $Ca^{2+}$ ions into the NSR from the SSL and Myo compartments. The formulation for $Ca^{2+}$ uptake via the SERCA pump consists of two terms. The first term is on the same form as $J_{RyR_{x_\infty}}$, so a greater $Ca^{2+}$ concentration in the compartment that the SERCA pump

pumps $Ca^{2+}$ from allows for a greater NSR uptake. The second term models the leak of $Ca^{2+}$ ions from the NSR. Writing $I_{leak}$ on the form:

$$I_{leak} = \frac{[Ca^{2+}]_{NSR}}{\tau \, \overline{NSR}}, \tag{3.28}$$

where $\tau$ [ms] is a time constant deciding the flux rate and $\overline{NSR}$ [dimensionless] is the maximum NSR $Ca^{2+}$ concentration, $I_{leak}$ is subtracted in the expression for $Ca^{2+}$ uptake via the SERCA pump. This means that the positive direction of the leakage current is opposite of that of $Ca^{2+}$ uptake via the SERCA pump. The flux $I_{leak}$ grows as a function of $[Ca^{2+}]_{NSR}$ and is inversely proportional to $\overline{NSR}$. This makes sense since a higher $Ca^{2+}$ concentration enables a higher leakage current, while a high maximum NSR $Ca^{2+}$ concentration means that more calcium can stay in the NSR and so the leakage current is lower than for a smaller maximum concentration.

The intracellular fluxes that are not involved in the SR are the results of concentration differences: a flux will go from areas with high concentration to areas with lower concentration. This is the form of the fluxes $J_{tr,j}$, $J_{tr,c}$, $J_{diff}$ and $J_{gap}$, that can be seen in the overview of the cell model in Figure 3.4, for example:

$$J_{tr,j} = \frac{[Ca^{2+}]_{NSR} - [Ca^{2+}]_{JSR}}{\tau_{tr}} \tag{3.29}$$

for $J_{tr,j}$, which is defined as the flux from the NSR to the JSR, and $\tau_{tr} = 120$ ms decides the flux rate.

### Ionic Concentrations

The PRd model has 11 ODEs describing ionic concentrations: one for $Ca^{2+}$ in each of the six compartments, for $Na^{+}$ in the Myo, SSL and PCS compartments, for $K^{+}$ in the Myo, and one for CAMKII. First ODEs are set up according to Figure 3.4, including all the fluxes. In this step it is important to scale for the different volumes that the fluxes are defined with respect to. The next step is to include $Ca^{2+}$ buffering.

For the first step, the $Ca^{2+}$ concentration in the SSL compartment will be used as an example. The equations for the other concentrations are found in the exact same manner. First all fluxes and currents to or from the SSL regarding $Ca^{2+}$ are identified from looking at the cell model figure: $I_{CaT}$, $I_{pCa}$, $I_{Cab}$ and $I_{NCX}$ across the cell membrane, $J_{diff}$ between the SSL and the PCS, $J_{gap}$ between the SSL and the Myo and $J_{SERCA,s}$ between the SSL and the NSR. Here $J_{diff}$ and $J_{gap}$ signifies the $Ca^{2+}$ part of the fluxes with the same names in the model overview in Figure 3.4, the total fluxes also have

a $Na^+$ part. The total amount of $Ca^{2+}$ contributed to the SSL compartment from the outside of the cell is then:

$$-(I_{CaT} + I_{pCa} + I_{Cab} - 2I_{NaCa,SSL})\frac{A_{cap}C_m}{z_{Ca}F}.$$

All the membrane currents have positive direction outwards by definition, therefore a positive membrane current contributes with a negative growth of the concentration, bringing ions out of the cell. This explains the minus sign in front of all the currents. However, the $Na^+$-$Ca^{2+}$ exchanger current $I_{NaCa,SSL}$ is defined as positive when it brings $Na^+$ ions out of the cell and $Ca^{2+}$ ions into the cell, increasing the intracellular $Ca^{2+}$ concentration, therefore an extra minus sign is needed to cancel the first one. The "2" in front of the $Na^+$-$Ca^{2+}$ exchanger-term is because $Ca^{2+}$-ions have a surplus of two positive charges. The naming in the original article is not completely consistent, but hopefully the meaning is clear from the context. Here $I_{NaCa,SSL}$ signifies the current throught the $Na^+$-$Ca^{2+}$ exchanger in the SSL compartment. In the overview in Figure 3.4 this is simply called $I_{NCX}$. To get the fluxes from the currents, the currents were divided by $z_{Ca}F$, and to get the total amount of $Ca^{2+}$, this was multiplied with the capacitive membrane area $A_{cap}$ [$cm^2$].

The flux $J_{diff}$ has positive direction from the PCS to the SSL, giving a plus sign, while $J_{gap}$ has positive direction from the SSL to the Myo, giving a minus sign, and $J_{SERCA,s}$ pumps $Ca^{2+}$ ions from the SSL into the NSR against the concentration gradient, giving a minus sign. All the fluxes are defined with respect to a reference volume [$\mu$L] (the smallest volume of the two compartments involved), and they are scaled by this volume in the equations, for example is $J_{SERCA,s}$ multiplied by the NSR volume. A form of scaling is necessary because a flux between two compartments will affect the concentration in the smallest volume the most. The scaling used here is quite standard for cell models.

These are the contributions to the change in the total amount of $Ca^{2+}$ in the SSL, $d(V_{SSL}[Ca^{2+}]_{SSL})/dt$. Since the model assumes that the compartmental volumes do not change, all terms can be divided by the SSL volume $V_{SSL}$, arriving at the final formulation:

$$\begin{aligned}\frac{d[Ca^{2+}]_{SSL}}{dt} = &-(I_{CaT} + I_{pCa} + I_{Cab} - 2I_{NaCa,SSL})\frac{A_{cap}C_m}{V_{SSL}\,z_{Ca}F} \\ &+ J_{diff}\frac{V_{PCS}}{V_{SSL}} - J_{SERCA,s}\frac{V_{NSR}}{V_{SSL}} - J_{gap} \quad (3.30)\end{aligned}$$

The next step is to include the effects of $Ca^{2+}$ buffers. Different types of such buffers are found in all compartments but the NSR. In these compart-

ments the total amount of $Ca^{2+}$ consists of the free amount and the buffered amount. The ODEs for the ionic concentrations model the free amount, but the free $Ca^{2+}$ is influenced by the buffered $Ca^{2+}$, therefore this must also be included in the model. Following Keener and Sneyd [14, eq.(7.69)], the basic chemical reaction for $Ca^{2+}$ buffering can be represented by:

$$P + Ca^{2+} \underset{k_-}{\overset{k_+}{\rightleftarrows}} B, \tag{3.31}$$

where P is the free buffering protein, "$Ca^{2+}$" is the free $Ca^{2+}$ and B is the complex of buffered $Ca^{2+}$. The variables $k_-$ and $k_+$ are a kind of rate constants. Letting $Buf$ [mmol/L] denote the concentration of buffer with $Ca^{2+}$ bound, corresponding to $B$, $[Ca^{2+}]$ be the concentration of free $Ca^{2+}$, and $\overline{Buf}$ be the total buffer concentration (which is assumed to be constant), corresponding to $P + B$, a simple model of $Ca^{2+}$ buffering is:

$$\frac{\partial [Ca^{2+}]}{\partial t} = D_c \nabla^2 [Ca^{2+}] + f([Ca^{2+}]) + k_- Buf - k_+ [Ca^{2+}](\overline{Buf} - Buf), \tag{3.32}$$

$$\frac{\partial Buf}{\partial t} = D_b \nabla^2 Buf - k_- Buf + k_+ [Ca^{2+}](\overline{Buf} - Buf). \tag{3.33}$$

Here $f([Ca^{2+}])$ denotes all the other reactions involving free $Ca^{2+}$. The variables $k_-$ and $k_+$ have units 1/ms and L/(mmol ms), respectively. The diffusion coefficients $D_c$ and $D_b$ have units $cm^2/ms$. Compartment can be denoted by a subscript. The buffer type $Buf$ is either calsequestrin (CSQN), troponin (TRPN) or calmodulin (CMDN). Since the compartmental model assumes constant concentrations within each compartment, the first term on the right hand side of both equations vanish. If the buffer reactions are fast compared to the other $Ca^{2+}$-reactions, $Buf$ can be assumed to be in quasi-steady state:

$$k_- Buf - k_+ [Ca^{2+}](\overline{Buf} - Buf) = 0, \tag{3.34}$$

which gives:

$$Buf = \overline{Buf} \frac{[Ca^{2+}]}{[Ca^{2+}] + K_{m,Buf}}, \tag{3.35}$$

where $K_{m,Buf} = k_-/k_+$ [mmol/L]. From here, two different approaches can be chosen. These will be explained consecutively. The first approach is as follows:

Assuming that the amount of free calcium at one time step is known, the buffered amount of calcium at the same time step can be found from equation

(3.35). From the ODEs on the form of equation (3.30) found earlier in this segment, an approximate value for the change in calcium during one time step, not including buffer effects, can be found. These three values are then added to approximate the total amount of calcium at the next time step. Now, keeping the values for the total amount of calcium and the change in calcium, and inserting the expressions for the free and buffered amount at the previous time step, one gets:

$$[Ca^{2+}]_{tot} = Buf + [Ca^{2+}] + d[Ca^{2+}] \tag{3.36}$$

where $[\mathrm{Ca}^{2+}]_{tot}$ and $\mathrm{d}[\mathrm{Ca}^{2+}]$ are known values, and $Buf$ and $[\mathrm{Ca}^{2+}]$ are unknown variables. Inserting the right hand side of equation (3.35) for $Buf$, this gives a polynomial equation with $[\mathrm{Ca}^{2+}]$ as the unknown. Solving the equation for the free amount gives a new approximate expression for the free calcium concentration where buffer effects are included. The solution will depend on $[\mathrm{Ca}^{2+}]_{tot}$ and $\mathrm{d}[\mathrm{Ca}^{2+}]$, so for each time step the ODEs must be solved to give $\mathrm{d}[\mathrm{Ca}^{2+}]$, while equation (3.35) must be used to find $Buf$.

The expression for the JSR and CSR compartments will have a single buffer term since they have only one buffer type, CSQN, while the expressions for the Myo, SSL and PCS compartments will have two buffer terms since they all have two types of calcium buffers, TRPN and CMDN:

$$[Ca^{2+}]_{tot} = CSQN + [Ca^{2+}] + d[Ca^{2+}], \tag{3.37}$$
$$[Ca^{2+}]_{tot} = TRPN + CMDN + [Ca^{2+}] + d[Ca^{2+}]. \tag{3.38}$$

For the JSR and the CSR compartments, both having a single calcium buffer, this yields a quadratic equation on the form:

$$(Ca)^2 + b(Ca) - c = 0, \tag{3.39}$$

with the free calcium as the unknown $Ca$ and

$$b = \overline{CSQN} - [Ca^{2+}]_{tot} + K_{m,CSQN},$$
$$c = K_{m,CSQN}[Ca^{2+}]_{tot}.$$

The quadratic equation (3.39) can be easily solved with the quadratic formula giving:

$$[Ca^{2+}] = \frac{-b \pm \sqrt{b^2 + 4c}}{2}. \tag{3.40}$$

Subtracting the square root will give a negative concentration, which does not make sense. Since the square root is greater than $b_x$, adding the square root will give a positive concentration, so this is the only valid solution.

The Myo, SSL and PCS compartments all have two types of calcium buffers, thus the given procedure results in a cubic equation for the amount of free calcium at the next time step:

$$(Ca)^3 + b(Ca)^2 + c(Ca) + d = 0, \tag{3.41}$$

with

$$\begin{aligned} b &= \overline{TRPN} + \overline{CMDN} - [Ca^{2+}]_{tot} + K_{m,TRPN} + K_{m,CMDN}, \\ c &= K_{m,TRPN}K_{m,CMDN} - [Ca^{2+}]_{tot}(K_{m,TRPN} + K_{m,CMDN}) \\ &\quad + \overline{TRPN}K_{m,CMDN} + \overline{CMDN}K_{m,TRPN}, \\ d &= -K_{m,TRPN}K_{m,CMDN}[Ca^{2+}]_{tot}. \end{aligned}$$

The cubic equation (3.41) can be solved analytically, assuming as before that $d[Ca^{2+}]$ is a number. The process of solving this equation is not shown here, but solution algorithms can be found (for example on wikipedia[1]) arriving at the solution through several steps with substitution of variables. The polynomial has three roots. As with the quadratic polynomial only one solution is valid in the context, the other two being either complex or negative. The solution is:

$$[Ca^{2+}] = \frac{2}{3}\sqrt{b^2 - 3c}\, cos\left(\frac{1}{3}\, cos^{-1}\left(\frac{9bc - 2b^3 - 27d}{2(b^2 - 3c)^{1.5}}\right)\right) - \frac{b}{3}. \tag{3.42}$$

Returning to equation (3.35) the second approach is described. This approach corresponds to adding together equations (3.32) and (3.33), with diffusion terms equal to zero, inserting the expression from equation (3.35) for $Buf$, and solving for $\partial[Ca^{2+}]/\partial t$. This yields an ODE for the amount of free calcium, including the buffer effects.

To get the exact same expressions as the ones that are used, a slightly different approach is chosen. Instead of adding the expressions for the time derivatives of $[Ca^{2+}]$ and $\overline{Buf}$, one defines:

$$v \equiv \overline{Buf} - Buf = \frac{\overline{Buf}K_{m,Buf}}{[Ca^{2+}] + K_{m,Buf}}, \tag{3.43}$$

and subtracts the time derivative of $v$ from the time derivative of $[Ca^{2+}]$:

$$\begin{aligned} \frac{\partial([Ca^{2+}] - v)}{\partial t} &= \frac{\partial}{\partial t}\left([Ca^{2+}] - \frac{\overline{Buf}K_{m,Buf}}{[Ca^{2+}] + K_{m,Buf}}\right) \\ &= \left(1 + \frac{\overline{Buf}K_{m,Buf}}{([Ca^{2+}] + K_{m,Buf})^2}\right)\frac{\partial[Ca^{2+}]}{\partial t} = f([Ca^{2+}]). \end{aligned}$$

[1] https://en.wikipedia.org/wiki/Cubic_function

Rearranging this gives:

$$\frac{\partial[Ca^{2+}]}{\partial t} = \left(1 + \frac{\overline{Buf}K_{m,Buf}}{([Ca^{2+}] + K_{m,Buf})^2}\right)^{-1} f([Ca^{2+}]) \equiv \beta f. \tag{3.44}$$

This expression only includes one buffer type. To include several buffer types, add a system on the form of (3.31) for each buffer. The variables $P$ and $B$ vary for each buffer, while $Ca^{2+}$ is the same in all reactions. For each additional buffer, the equation for the time derivative of $[Ca^{2+}]$ must be modified by adding two terms on the form $k_- Buf - k_+[Ca^{2+}](\overline{Buf} - Buf)$, and an extra equation on the form of equation (3.33) must be added to the system. Each buffer is assumed to be in quasi-steady state, giving an expression on the form of equation (3.35) for each buffer. A $v$ is defined in the same manner as before, and all $v$'s are subtracted from the time derivative of $[Ca^{2+}]$. In the end, each buffer type contributes to $\beta$ with a term on the form:

$$\theta_{Buf} = \frac{\overline{Buf}K_{m,Buf}}{([Ca^{2+}] + K_{m,Buf})^2}, \tag{3.45}$$

yielding:

$$\beta = \frac{1}{1 + \sum_{Buf} \theta_{Buf}}. \tag{3.46}$$

This formulation is used for the PCS compartment, while the other compartments use the first buffer approach. However, all the buffer equations had to be formulated this way to be defined in gotran format.

### CAMKII.

CAMKII stands for $Ca^{2+}$/calmodulin-dependent protein kinase II. Calmodulin is short for $Ca^{2+}$-modulated protein. It is a protein binding $Ca^{2+}$-ions and other proteins. Many of these other proteins are not able to bind $Ca^{2+}$ themselves, and use calmodulin as a $Ca^{2+}$ sensor. A kinase is a type of enzyme catalyzing phosphorylation, which is the addition of a phosphate group to a protein. This can turn protein enzymes on or off, changing the function and activity of the protein. CAMKII is then a type of enzyme regulating the activity of a protein, and this enzyme again depends on the $Ca^{2+}$/calmodulin-complex. There are several groups of such enzymes, II refers to the specific group.

CAMKII plays an important role in regulating the release and uptake of $Ca^{2+}$, and thus in controlling the intracellular $Ca^{2+}$-concentration. It is included in the ODEs for the fluxes $J_{RyR_2}$ and $J_{RyR_3}$, where it plays a role

in regulating $Ca^{2+}$ release via the Ryanodine receptor (RyR) channels, it is taken into account in the expressions for the SERCA-pumps, regulating the uptake of $Ca^{2+}$ to the SR, and finally, it is found in the expression for the $I_{CaL}$ current, regulating the transmembrane inflow of $Ca^{2+}$ to the PCS compartment.

### 3.1.4 The Decker et al. Model

The Decker et al. cell model models an epicardial myocyte. This model is used for all the ventricular muscle cells. For the full model the reader is referred to the original publication [15], but an overview will be presented here. The cell model consists of 46 coupled ODEs, lots of expressions and 85 parameters. The state vector consists of the transmembrane potential, 11 ionic concentrations, 23 states, 10 gating variables and one flux. The state variables can be seen in Table 3.2.

| | | | |
|---|---|---|---|
| $V$ | $[K^+]_i$ | $C^*$ | $m_L$ |
| $[Ca^{2+}]_{SS,CaL}$ | $[Cl^-]_{SS,SR}$ | $O^*$ | $h_L$ |
| $[Ca^{2+}]_{SS,SR}$ | $[Cl^-]_i$ | $CI^*$ | $X_r$ |
| $[Ca^{2+}]_{JSR}$ | $[CAMK]_{trap}$ | $C_1 - C_{15}$ | $a$ |
| $[Ca^{2+}]_{NSR}$ | $C$ | $O_1$ | $i_{1f}$ |
| $[Ca^{2+}]_i$ | $O$ | $m$ | $i_{1s}$ |
| $[Na^+]_{SS,SR}$ | $CI$ | $h$ | $i_{2f}$ |
| $[Na^+]_i$ | $OI$ | $j$ | $I_{rel}$ |

**Table 3.2:** State variables in the Decker et al. model. $V$ is the transmembrane potential, $[X]_y$ is the concentration of X in compartment y, the variables from $C$ to $O_1$ are states in Markov models and the rest are gating variables in the expressions for the ionic currents.

Many of the model equations and expressions are on similar form as in the PRd model but with different parameter values. The equation for the transmembrane potential is on the same form as in the PRd model, see equation (3.5), where $C_m$ is incorporated into the expressions for the transmembrane currents, which then have units $\mu A/\mu F$. The action potential can be seen in Figure 3.5.

The ionic currents have the same form as in equations (3.25) and (3.26), but the collective open probability $O$ of the subunits of an ion channel is modeled in two ways: by gating variables, as in the PRd model, and by Markov models, which will be further explained below.

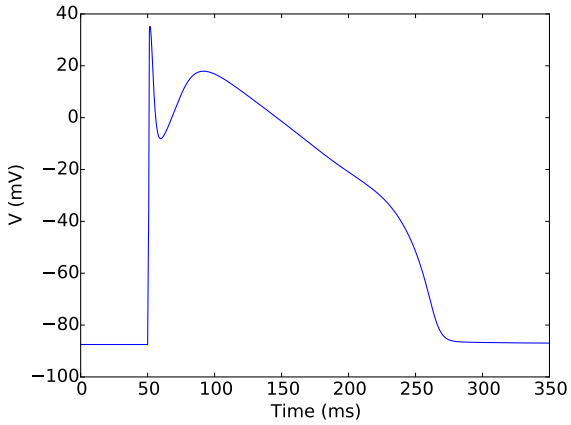



**Figure 3.5:** Action potential for a ventricular myocyte with the Decker et al. cell model.

The expressions for the ionic concentrations are found in the same way as in the PRd model and will not be further explained here. For those compartments containing $Ca^{2+}$ buffers the formulations were either on ODE form or solutions of second and third order polynomial equations, as for the PRd model. However, in the CellML code for the model [23] these were all formulated as ODEs, so no modifications were necessary to get these equations on gotran form. In addition to the pumps and exchangers found in the PRd model, the Decker et al. model also have $Na^+$-$Cl^-$- and $K^+$-$Cl^-$ cotransporters.

CAMKII was described in the PRd model. In the Decker et al. model it is included in the expressions for SR $Ca^{2+}$ release via the $I_{Rel}$ current and uptake via the $I_{up}$ current (this uptake is through a SERCA-pump). In addition the effects from CAMKII is included in the Markov model for the $I_{CaL}$ current.

### Cell Structure

The Decker et al. cell model is a compartmental model with five compartments: the myoplasma (MYOPLASM), the network sarcoplasmic reticulum (NSR), the junctional sarcoplasmic reticulum (JSR), the sarcoplasmic reticulum subspace (SS(SR)) and the $I_{CaL}$ subspace (SS(CaL)). An illustration of the model can be seen in Figure 3.6.

The cell myoplasma is divided into three compartments: SS(CaL) is a small compartment where the L-type $Ca^{2+}$ channels are placed, SS(SR) is another small compartment into which $Ca^{2+}$ ions are released from the SR,

and the rest is MYOPLASM, which contains most of the transmembrane ion channels. This last compartment also has two types of buffers: calmodulin (CMDN) and troponin (TRPN).

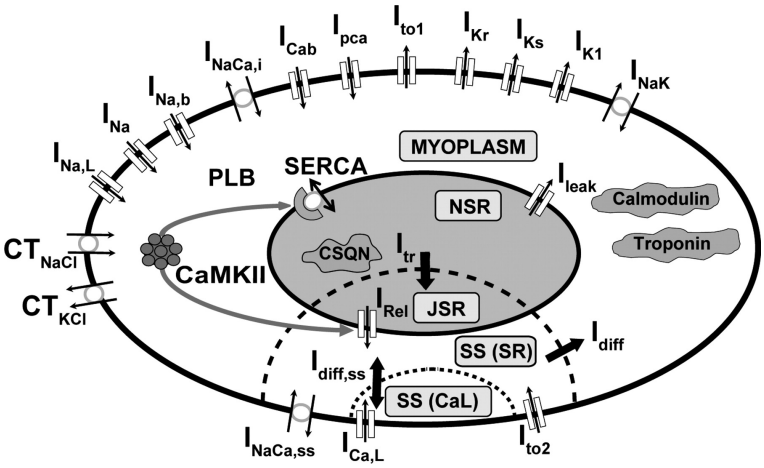



**Figure 3.6:** A graphic overview of the Decker et al. cell model for a canine epicardial myocyte. This is Figure 1 from page H1018 in the article "Properties and ionic mechanisms of action potential adaptation, restitution, and accomodation in canine epicardium", by authors Keith F. Decker, Jordi Heijman, Jonathan R. Silva, Thomas J. Hund and Yoram Rudy, published in *American Journal of Physiology - Heart and Circulatory Physiology* 2009, Vol. 296, no. 4, H1017-H1026 [15]. Reproduced with permission.

The SR is divided into two compartments: the NSR takes up calcium from the myoplasma, from here it diffuses into the JSR, which release calcium back into the myoplasma when triggered. The JSR contains a calcium buffer called calsequestrin (CSQN). The different transmembrane currents and intracellular fluxes can be seen in the overview of the cell in Figure 3.6.

### Markov Models

The Decker et al. cell model uses Markov models to find the open probabilities in the $I_{CaL}$ and $I_{Ks}$ currents. A Markov model is a stochastic model used to model systems changing randomly between states, where it is assumed that future states depend only on the present state. In cell models the Markov models model the states of single ion channels. While the Hodgkin-Huxley formulation assumes independent gating for each ion channel, which is typically not the case, the advantage of Markov models is that they enable

more specific modeling of each ion channel, including dependencies between the states. In the case of independent gating the Markov model and the Hodgkin-Huxley formulation are equivalent (based on lecture notes from the course INF5610 taught at the University of Oslo [19]).

The main state conformations in a cell model are the open and closed states. If a channel consists of several subunits, there are several closed states and the open state signifies that all subunits are open. In addition there is inactivation. This can be treated as one state, or it can be split into open inactivated and closed inactivated, which allows for different transition rates for these two cases. When an ion channel is in the open activated state it can conduct current, otherwise not. The inactivated property accounts for the refractory period of an action potential. If a channel has been in the open state, leading to the depolarization of the cell, and then transitions directly into a closed state, it can be opened again without delay, depending on the membrane potential. Instead it transitions from the open activated state into an inactivated state where no stimulus can trigger a new depolarization. After a while it transitions from the inactivated state to the closed activated state, ready for a new action potential.

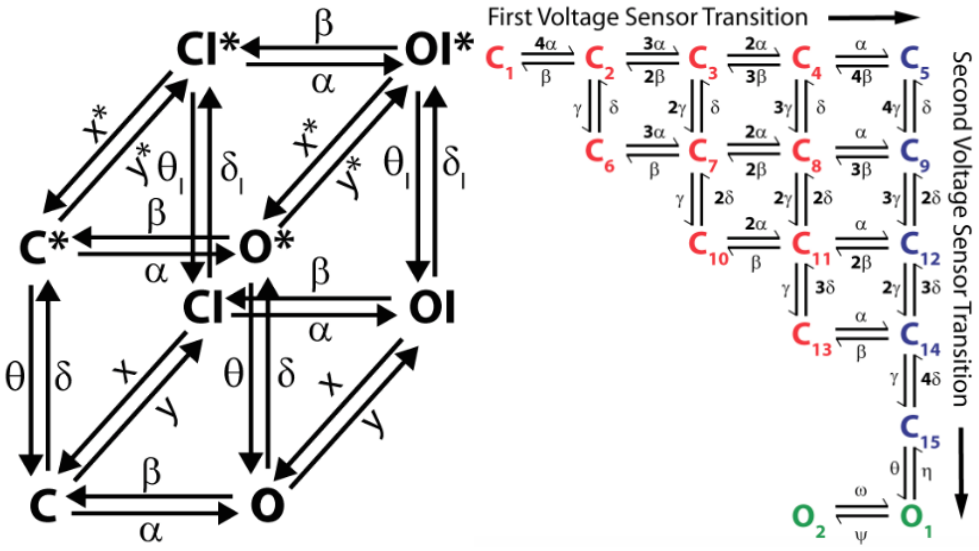
Transition rates decide the probability of transition from one state to another. Together the states and transition rates make up the complete model, which can be expressed in a Markov state diagram. The diagrams for $I_{CaL}$ and $I_{Ks}$ are shown in Figure 3.7.

The Markov model for $I_{CaL}$ in the Decker et al. model is new. It has four open and four closed states, as can be seen in Figure 3.7a. From the figure one can set up the ODE governing each state. This gives a system of coupled ODEs. For $O^*$ for example, there are three states that can move into state $O^*$, and state $O^*$ can also move into these three states. The change in time of a state's "concentration" is equal to the sum of all possible transitions. Each transition is proportional to the concentration of the state that is being left, and the transition rates are the proportionality constants:

$$\frac{dO^*}{dt} = -(\beta + \theta + y^*)\, O^* + \delta\, O + \alpha\, C^* + x^*\, O I^*. \tag{3.47}$$

The probability used in equations (3.25) or (3.26) for the transmembrane current, replacing the gating probability $O$ in equation (3.15), is $O + O^*$, where $O$ here is a state in the Markov model, as seen in Figure 3.7a. The transition rates depend on the transmembrane current $V$ and the calcium concentration in the SS(CaL) compartment $[Ca^{2+}]_{SS,CaL}$.

The Markov model for $I_{Ks}$ has adjusted parameters from the original model by Silva and Rudy (see reference in Decker et al. [15]), to fit data for canine myocytes. It has 17 states, 15 closed and two open, as shown




**(a)** $I_{CaL}$ **(b)** $I_{Ks}$

**Figure 3.7:** Markov models for the (a) $I_{CaL}$ and (b) $I_{Ks}$ currents. (a) and (b) are Figures S1 and S2 from pages 7 and 10, respectively, in the online supplement to the article "Properties and ionic mechanisms of action potential adaptation, restitution, and accomodation in canine epicardium", by Decker et al., published in *American Journal of Physiology - Heart and Circulatory Physiology* 2009, Vol. 296, no. 4, H1017-H1026 [15]. Reproduced with permission.

in Figure 3.7b. The ODEs are set up as before, and solved, yielding the open probability $O = O_1 + O_2$. The transition rates depend only on the transmembrane current $V$.

### Ionic Currents

Four types of ions are included in the Decker et al. model: sodium, potassium, calcium and chloride. The total transmembrane current is described as a sum of the currents of these individual ions, so equation (3.5) becomes:

$$\frac{dV}{dt} = -(I_{Na,tot} + I_{K,tot} + I_{Ca,tot} + I_{Cl,tot}). \tag{3.48}$$

Each of these currents are again described as sums of different types of currents of the same ion, in the same manner as for the PRd cell model. The

equations for the total flow of the different ion types are:

$$I_{Na,tot} = I_{Na} + I_{NaL} + I_{Nab} + 3I_{NaK} + 3(I_{NaCa} + I_{NaCa,ss}), \tag{3.49}$$
$$I_{K,tot} = I_{Kr} + I_{Ks} + I_{K1} + I_{Kp} - 2I_{NaK} + I_{to1}, \tag{3.50}$$
$$I_{Ca,tot} = I_{CaL} + I_{Cab} + I_{pCa} - 2(I_{NaCa} + I_{NaCa,ss}), \tag{3.51}$$
$$I_{Cl,tot} = I_{Clb} + I_{to2}. \tag{3.52}$$

These currents follow the same rules as those of the PRd cell model.

#### SR $Ca^{2+}$ Fluxes

The model for $Ca^{2+}$ fluxes in the Decker et al. model differs from that of the PRd model since the compartments are different. This model has only one flux that releases $Ca^{2+}$ from the SR. It is called $I_{rel}$:

$$I_{rel} = -\frac{I_{rel,\infty} + I_{rel}}{I_{rel,\tau}}. \tag{3.53}$$

The form differ slightly from that of the $J_{RyR}$ fluxes in the PRd model, having a minus sign in front of $I_{rel,\infty}$, but similar to the PRd case, $I_{rel}$ is a function of variables $I_{rel,\tau}$ and $I_{rel,\infty}$. The variable $I_{rel,\tau}$ is on the same form as $\tau_{RyR_x}$. The expression for $I_{rel,\infty}$ is similar to the expression for $RyR_{x_\infty}$, but it is not on the exact same form. The difference in form accounts for the opposite signs.

The remaining expressions for SR $Ca^{2+}$ fluxes are on the same form as in the PRd model, only with different parameters, and what is termed $Ca^{2+}$ uptake via SERCA in the PRd model is split in two in the Decker et al. model: SR uptake ($I_{up}$, corresponds to the first term in $J_{SERCA}$) and SR leak ($I_{leak}$).

### 3.1.5 Comparing the PRd and the Decker et al. Models

From sections 2.4.3, 3.1.3 and 3.1.4 it is known that the PRd and the Decker et al. cell models model the same currents with only a few exceptions. The phases and roles of the individual currents during an action potential were treated in Section 2.4.3. Still, the action potentials of these two cell types are quite different, as can be seen in Figure 3.8.

The shapes have marked differences, most notably the action potential duration and presence, or lack thereof, of a phase 1 notch. They both have sharp phase 0 upstrokes, but while the membrane potential of the Purkinje cell falls monotonically from its peak, faster during phases 1 and 3 than during the plateau phase, the myocyte potential has a deep notch (more than

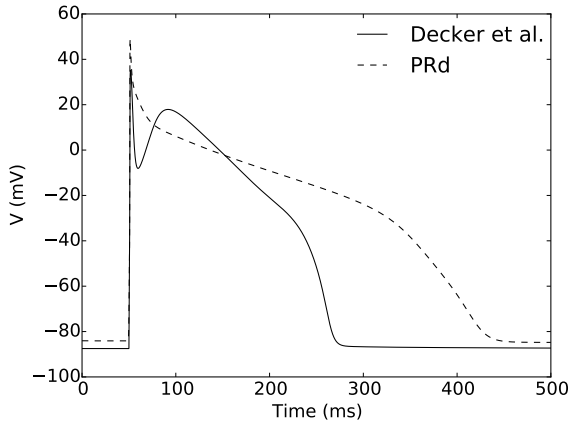



**Figure 3.8:** Action potentials for the PRd and the Decker et al. cell models. The five phases were described in Section 2.4.3, see Figure 2.3: 0:depolarization, 1:first part of repolarization/notch, 2:plateau, 3:rapid repolarization, 4:diastolic depolarization/resting state.

20 mV) during phase 1, before the plateau phase, where it falls considerably faster than the Purkinje potential, finally diving steeply during phase 3. The action potential duration of the ventricular myocyte of the Decker et al. model is about 220 ms, which is much shorter than the about 380 ms for the Purkinje cell of the PRd model.

An important current regarding the action potential duration is $I_{NaL}$. If this current is removed, the action potential duration for the Decker et al. model reduces from about 220 ms to about 190 ms, while the action potential duration for the PRd model reduces from about 380 ms to 190 ms, a much more dramatic change. The greater effect for the PRd model comes from a higher expression level of $I_{NaL}$ in this model than in the Decker et al. model.

## 3.2 Cardiac Tissue Model

### 3.2.1 The Monodomain Equation

The monodomain equation is a simplification of the bidomain equations, which is a mathematical model for electrical signal propagation in the cardiac tissue. The basic assumption of the bidomain model is that the heart consists of two separate, continuous domains: the intracellular and extracellular spaces, separated by the cell membrane. Viewing the intracellular space as continuous is quite reasonable because of the gap junctions. It is a continuous model, using a volume-averaging approach. This means that instead of modeling the individual cells of the tissue, the tissue is viewed as continuous: a quantity at a point is modeled as the average over a small but multicellular volume around the point.

The heart tissue is modeled as a volume conductor. One of Maxwell's equations is:

$$\nabla \times E + \frac{\partial B}{\partial t} = 0, \tag{3.54}$$

where $E$ [mV/cm] and $B$ [N/($10^{10}$ $\mu$A cm)] are the strengths of the electric and magnetic fields, respectively. For the heart it is reasonable to treat the fields as static for each moment in time. This is called the quasi-static condition. Then equation (3.54) reduces to:

$$\nabla \times E = 0. \tag{3.55}$$

This again ensures that $E$ can be written as the gradient of a scalar-valued potential $u$ [mV]:

$$E = -\nabla u. \tag{3.56}$$

The current $J$ [$\mu$A/cm$^2$] in a conductor is given as the product of the conductivity $M$ [mS/cm] of the tissue and the electrical field $E$, giving:

$$J = ME = -M\nabla u. \tag{3.57}$$

The quasi-static assumption demands conservation of current, so the current leaving the extracellular space in a point must be equal in size and opposite in direction to the current leaving the intracellular space in the same point:

$$\nabla \cdot J_i = -\chi I_m, \tag{3.58}$$
$$\nabla \cdot J_e = \chi I_m. \tag{3.59}$$

The transmembrane current $I_m$ [$\mu$A/cm$^2$] is per unit area of membrane surface. The variable $\chi$ [cm$^{-1}$] is the amount of membrane surface area per

unit volume of tissue. Thus multiplication by $\chi$ scales the current to be per unit volume, which is in accordance with the divergence on the left hand sides.

Using equation (3.57) for $J_i$ and $J_e$, replacing $V_i$ by $V + V_e$ and inserting for $I_m$ from equation (3.4), equations (3.58) and (3.59) can be written as equations (3.60) and (3.61), which is the standard formulation of the bidomain equations. The first equation follows directly from equation (3.58), while the second results from adding the two equations:

$$\nabla \cdot (M_i \nabla V) + \nabla \cdot (M_i \nabla V_e) = \chi C_m \frac{\partial V}{\partial t} + \chi I_{ion}, \tag{3.60}$$

$$\nabla \cdot (M_i \nabla V) + \nabla \cdot ((M_i + M_e) \nabla V_e) = 0. \tag{3.61}$$

Here $C_m$ is the capacitance of the cell membrane and $I_{ion}$ is the resistive current carrying ions across the cell membrane, as described in Section 3.1.

As explained in Section 2.2.2 the heart tissue is strongly anisotropic, making the conductivities $M_i$ and $M_e$ tensor quantities. By assuming a simplification of the conductive properties of the heart, the bidomain equations can be transformed into the monodomain equation. The intracellular and extracellular conductivities are different, but choosing $M_e = \lambda M_i$, where $\lambda$ is a constant scalar, the bidomain model can be reduced to the monodomain model.

Replacing $M_e$ in equation (3.61) by $\lambda M_i$ yields:

$$\nabla \cdot (M_i \nabla V_e) = -\frac{1}{1+\lambda} \nabla \cdot (M_i \nabla V). \tag{3.62}$$

Inserting this into equation (3.60) and rearranging gives the standard formulation of the monodomain model (Sundnes et al. [24, eq.(2.35)]):

$$\frac{\lambda}{1+\lambda} \nabla \cdot (M_i \nabla V) = \chi C_m \frac{\partial V}{\partial t} + \chi I_{ion}. \tag{3.63}$$

Equating the right hand side of equation (3.63) with zero, equation (3.4) for the membrane potential in a single cell model will no applied stimulus current can be recognized. The extra term adds spatial diffusion since it is now a tissue that is being considered, and the result is a reaction-diffusion PDE.

The boundary condition is given as:

$$n \cdot (M_i \nabla V) = 0, \tag{3.64}$$

but this will be treated more closely in Section 4.3.

Fiber directions in the myocardial tissue are not included in this thesis, so $M_i$ is assumed to be spatially independent and can be drawn outside the divergence operator. Deo et al. [6] define the intracellular conductivity in the tissue by $\sigma = M_i \frac{\lambda}{1+\lambda}$ [mS/cm]. In this thesis the monodomain equation is rewritten to contain a diffusion coefficient, as this will be practical for implementational purposes. The diffusivity $D$ [cm$^2$/ms] of the tissue is found by dividing the conductivity $\sigma$ by $\chi C_m$: $D = \frac{\sigma}{\chi C_m}$. Now, incorporating $C_m$ into $I_{ion}$, as described in the beginning of Section 3.1, using $D$ and rearranging the terms, the monodomain equation can be written as:

$$\frac{\partial V}{\partial t} = D\Delta V + I_{ion}(s), \tag{3.65}$$

where $s$ signifies the state vector of the cell model (this includes the membrane potential). This equation is quite similar to equation (3.5) except it has a spatial term not present in equation (3.5). Since $I_{ion}$ depends on a number of variables, collectively labeled $s$, it is necessary to solve a set of equations on the form

$$\frac{\partial s}{\partial t} = f(s, t) \tag{3.66}$$

for each $s$, except that the equation governing the membrane potential $V$ is now replaced by the monodomain equation. So the complete monodomain model is given by equations (3.65) and (3.66), where the first equation (the monodomain equation) also needs a boundary condition. This gives a coupled system of nonlinear ODEs, as for the single cell models, but in addition there is also a nonlinear PDE. In general, there is no analytical solution to this problem, and no simple numerical solution either. To simplify the problem, operator splitting was used on the monodomain equation.

# Chapter 4

# Numerical Methods

The PRd [17] and Decker et al. [15] cell models and the monodomain equation, combined with sufficient initial conditions, hold all the information necessary to simulate the electrical activity in the ventricles. To get access to this information, the differential equations must be solved. This can be a very challenging task. Only some types of differential equations have known analytical solutions. Not surprisingly, the equations in this problem are not among them, therefore it is necessary to turn to numerical methods. To solve these equations numerically one must first split the problem into simpler parts, and then solve each part using known methods. This procedure is described in this chapter.

## 4.1 Time Discretization

The original problem is continuous both in time and space, but to be able to solve the equations numerically they must be restricted to discrete times and spatial points. Uniformly spaced time points are used for the time discretization. If the problem is to be solved for the time interval $[0, T]$, at $N + 1$ times, where $N$ can be fitted to the required resolution, the times are $t_n = n\Delta t$, $n = 0, 1, 2, ..., N$, where $\Delta t = T/N$. The approximate solution at time $t_n$ is denoted by the superscript $n$, for example $V^n$.

### 4.1.1 Operator Splitting

Operator splitting is a technique for solving coupled systems of PDEs by splitting the system into smaller parts, each of which can be solved more easily than the complete system. Here the first-order method of Godunov splitting is used, as described in Sundnes et al. [24, sections 3.2.1 and 3.2.2].

For a general problem on the form:

$$\frac{\partial u}{\partial t} = (L_1 + L_2)u, \quad u(0) = u_0, \tag{4.1}$$

where $L_1$ and $L_2$ are operators on $u$, and $u_0$ is the initial condition, an approximation of the solution at $t = \Delta t$ can be found by first solving the problem:

$$\frac{\partial v}{\partial t} = L_1 v, \quad v(0) = u_0 \tag{4.2}$$

for $t \in [0, \Delta t]$, then solving the problem:

$$\frac{\partial w}{\partial t} = L_2 w, \quad w(0) = v(\Delta t) \tag{4.3}$$

for $t \in [0, \Delta t]$.

A first-order approximation to the true solution $u$ at $t = \Delta t$ is given by $w(\Delta t)$. For the monodomain equation on the form (3.65), one can define the operators $L_1$ and $L_2$ in the following way:

$$L_1 V = I_{ion}(s), \tag{4.4}$$
$$L_2 V = D\Delta V. \tag{4.5}$$

If $V^n = V(t_n)$ and $s^n = s(t_n)$ are known solutions to the monodomain equation and equation (3.66) at time $t = t_n$, an approximation at $t = t_n + \Delta t$ can be found by first solving the system of nonlinear ODEs:

$$\frac{\partial V}{\partial t} = I_{ion}(s), \quad V(t_n) = V^n, \tag{4.6}$$
$$\frac{\partial s}{\partial t} = f(s), \quad s(t_n) = s^n, \tag{4.7}$$

for $t_n < t < t_n + \Delta t$ (where the equation for $V$ is excepted from the second equation). Since the diffusion term is removed from the monodomain equation in this first step, the equation for $V$ is reduced to the equation for $V$ in the single cell models, so the complete step corresponds to solving the coupled system of ODEs for a single cell. These equations were described earlier in this chapter.

The next step is to solve the linear PDE:

$$\frac{\partial V}{\partial t} = D\Delta V, \quad V(t_n) = V_*^n \tag{4.8}$$

for $t_n < t < t_n + \Delta t$, where $V_*^n$ is the solution yielded by solving (4.6) in the previous step. This procedure is repeated for each time step.

### 4.1.2 The Finite Difference Method

Differences between solutions at different time steps are used to approximate the time derivative, hence the name of the method. There are many different approximation schemes, of varying order.

#### FHN Model

The equations in the FHN cell model are repeated here for convenience:

$$\epsilon \frac{dV}{dt} = f(V) - w + I_{app}, \tag{4.9}$$

$$\frac{dw}{dt} = V - \gamma w. \tag{4.10}$$

Since this is a test problem, the simplest finite difference method is used, a forward Euler sceme:

$$\frac{\partial V^n}{\partial t} \approx \frac{V^{n+1} - V^n}{\Delta t}, \tag{4.11}$$

and the rest of the terms are at time step $n$.

This results in an explicit scheme, which is very unstable with respect to the time resolution, but as the problem is quite small (only two ODEs), and an explicit scheme is fast to solve, this is remedied simply by refining the time step enough to give a stable solution. The resulting scheme is first order, so the error is proportional to $\Delta t$. The complete scheme is:

$$V^{n+1} = V^n + \frac{\Delta t}{\epsilon}(f(V^n) - w^n + I_{app}), \tag{4.12}$$

$$w^{n+1} = w^n + \Delta t(V^{n+1} - \gamma w). \tag{4.13}$$

#### Monodomain Equation

Since this is the main problem it is easy to think that the best solution is to use a very high order scheme, so the error will be as small as possible. This is of course correct in one respect, but the higher order the scheme, the more demanding it will be to solve. Therefore it is necessary to find the balance between time consumption to find the solution, and the required accuracy. The other aspect to bear in mind is that the accuracy of the approximate solution is limited by the lowest order scheme that is used to find it, to use higher order schemes for parts of the problem will only be a waste. The numerical scheme used for operator splitting on the monodomain problem was first order, therefore a first-order method is also chosen for the other time discretization. There are two natural choices for first order finite

difference schemes: forward Euler and backward Euler. Since the forward Euler scheme can be unstable, depending on the time step, a backward Euler scheme is chosen:

$$\frac{\partial V^n}{\partial t} \approx \frac{V^n - V^{n-1}}{\Delta t}. \tag{4.14}$$

The nonlinear PDE resulting from operator splitting on the monodomain equation is:

$$\frac{\partial V}{\partial t} = D\Delta V \tag{4.15}$$

Using the scheme in equation (4.14) on this gives:

$$\frac{V^{n+1} - V^n}{\Delta t} = D\Delta V^{n+1}. \tag{4.16}$$

## 4.2 Spatial Discretization

The finite element method is used to solve PDEs. In this method the area on which the problem is to be solved, called $\Omega$, is approximated by many small, non-overlapping areas called elements. In 1D these elements are intervals, in 2D they are surfaces and in 3D they are volumes. Here triangles are used in 2D and tetrahedra in 3D. Together the elements make up a polygonal approximation of the original area. This polygonal approximation is labeled $\Omega_h$, and as the element size decreases, it will approach $\Omega$. $\Omega_h$ is called a mesh and $\Omega_h \subset \Omega$, with equality if $\Omega$ is a polygonal area and $\Omega_h$ has enough elements.

A facet has dimension $D - 1$. In 1D it is an end point of an interval element, in 2D it is an interval edge of a surface element and in 3D it is a surface on a volume element.

The vertices, also called nodes, are the grid points in the mesh $\Omega_h$. The elements are drawn up between the vertices, and the boundary vertices in $\Omega_h$ are on the boundary of $\Omega$, $\partial\Omega$. For each vertex in the mesh grid one defines a piecewise polynomial basis function $\phi$. Basis function number $i$ is defined to have the value one at grid point $i$ and zero at all other grid points:

$$\phi_i(x_j) = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

The polygonal order of the basis functions decide the order of accuracy of the finite element method. The lowest order functions giving a continuous solution are linear elements. This gives second order accuracy (in space) and was chosen here.

The solution of the weak formulation (see Section 4.2.1) of the problem is in a suitably defined function space $H^1$ defined over the domain $\Omega$. Through the finite element method a finite dimensional approximation to this solution is found in the function space $H_h^1$ of dimension $n$, defined over the domain $\Omega_h$. The function space $H_h^1$ is the space that has $\phi_i, \quad i = 1, ..., n$ as basis functions, and any function $u_h \in H_h^1$ can be written as a linear combination of these functions:

$$u_h = \sum_{j=1}^{n} \alpha_j \phi_j, \tag{4.18}$$

where the $\alpha$'s are scalars. $H_h^1 \subset H^1$.

### 4.2.1 Weak Form

The weak form, or variational formulation, of a PDE is found by multiplying the equation by a test function $\psi \in V$ and integrating over the whole domain $\Omega$. Operator splitting on the monodomain equation resulted in a linear PDE (eq.(4.8)):

$$V_t = D\Delta V. \tag{4.19}$$

From this one gets:

$$\int_\Omega V_t \, \psi \, dx = \int_\Omega D\Delta V \, \psi \, dx. \tag{4.20}$$

This is required to be fulfilled for all $\psi \in H^1$. Independence of space has already been assumed for $D$, so it can be taken outside the integral. Applying Green's lemma to the right hand side and rearranging the terms, one arrives at the weak form:

$$\int_\Omega V_t \, \psi \, dx - D \int_{\partial\Omega} (\nabla V \cdot n) \, \psi \, ds + D \int_\Omega \nabla V \cdot \nabla \psi \, dx = 0 \quad \forall \, \psi \in H^1. \tag{4.21}$$

This form is weaker than the original formulation in two respects. First, the original formulation required that the PDE was satisfied for all points in $\Omega$, while the weak form requires that the integral equation of the weak form holds for all $\psi$ in $H^1$. Since the integral represents a form of averaging, this second requirement is weaker than the original one. Secondly, the weak form only involves first derivatives of $V$, while the original PDE has a second derivative, which imposes stricter smoothness requirements on $V$. It can be proven that any solution of the weak form that is twice differentiable is also a solution of the original PDE.

The 1D version of equation (4.21) is obtained by the 1D version of Green's lemma. Since this equation is easier to work with and the mesh for the Purkinje tree will be in 1D, this version is also presented explicitly:

$$\int_\Omega V_t\,\psi\,dx - D\,V_x(B)\,\psi(B) + D\,V_x(A)\,\psi(A) + D\int_\Omega V_x\,\psi_x\,dx = 0 \quad \forall\ \psi \in H^1, \tag{4.22}$$

where $\Omega$ is the interval from $A$ to $B$.

The finite element method solves the discrete version of equation (4.21):

$$\int_{\Omega_h} U_t\,\psi\,dx - D\int_{\partial\Omega_h} (\nabla U\cdot n)\,\psi\,ds + D\int_{\Omega_h} \nabla U\cdot\nabla\psi\,dx = 0 \quad \forall\ \psi \in H_h^1, \tag{4.23}$$

where $U$ approximates $V$. In the following, $U$ will not be kept as notation, and the real and approximate solution will not be notationally distinguished, but which solution is relevant should be clear from the context.

Using the time discretization introduced in the previous section on $U_t$ one gets:

$$\int_{\Omega_h} \frac{V^{n+1}-V^n}{\Delta t}\,\psi\,dx - D\int_{\partial\Omega_h} (\nabla V^{n+1}\cdot n)\,\psi\,ds + D\int_{\Omega_h} \nabla V^{n+1}\cdot\nabla\psi\,dx = 0, \tag{4.24}$$

where $V^{n+1}$ is unknown. Since $V^{n+1} \in H_h^1$, it can be written as:

$$V^{n+1} = \sum_{j=1}^{n} v_j\phi_j, \tag{4.25}$$

where the $v$'s are unknown scalars. If these are found, $V^{n+1}$ is known.

Equation (4.24) must hold for all $\psi \in H_h^1$. This is fulfilled if it holds for all basis functions of $H_h^1$, i.e., if it holds for $\psi = \phi_i$ for all $i$. Inserting equation (4.25) for $V^{n+1}$ gives $n$ linear equations with the $n$ $v$'s as unknows. This is solvable.

Neumann boundary conditions appear naturally in the formulation as a boundary integral, and are called natural boundary conditions, while Dirichlet boundary conditions must be forced, and are called essential boundary conditions.

### 4.2.2 One Equation on the Whole Purkinje Domain

An illustration of the Purkinje network used in this thesis can be seen in Figure 4.1.

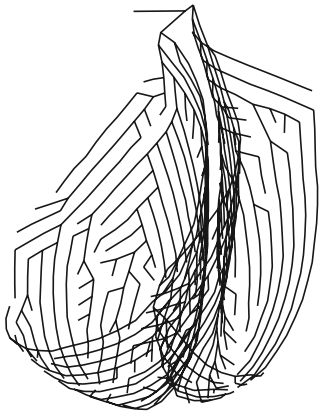


**Figure 4.1:** Purkinje network

The branches in the Purkinje network are described by the 1D monodomain model:

$$v_t = Dv_{xx} + I_{ion}, \tag{4.26}$$

where $v$ is the transmembrane potential. There is one such equation for each segment $e_i$. The weak form resulting from this (where the $I_{ion}$ term is removed because of operator splitting) is given in equation (4.22). On the junctions between branches there are made two assumptions: continuity of potentials and conservation of current. For a general junction $J$ which joins a set of $n$ segments $e_i, i = 1, \ldots n$ with solutions $v_i$ and conductivity $\sigma_i$, the following hold:

$$v_i = v_j \quad \forall\ i, j, \tag{4.27}$$

$$\sum_i \sigma_i (v_i)_x = 0. \tag{4.28}$$

Summing the equations for all the branches, which are all on the form of equation (4.22), the boundary terms become:

$$\sum_i \left( -D_i \left(v_i\right)_x (B_i)\, \psi_i(B_i) + D_i \left(v_i\right)_x (A_i)\, \psi_i(A_i) \right). \tag{4.29}$$

In Section 3.2.1, the diffusion coefficient $D$ was defined as the conductivity $\sigma$ divided by $\chi C_m$: $D = \sigma/(\chi C_m)$. Since all the branches are built up of

Purkinje cells, $\chi$ and $C_m$ are the same on all the branches, so equation (4.28) can be replaced by:

$$\sum_i D_i (v_i)_x = 0, \tag{4.30}$$

where $D_i = \sigma_i/(\chi C_m)$.

All boundary terms at junctions vanish because of condition (4.30) and because it is possible to choose local test functions $\psi$ such that $\psi_i(J) = \psi_j(J)$, where $J$ is the junction between two or more branches, and thus a common facet on adjacent elements. (This is not fulfilled for all choices of local test functions, but it is always possible to choose test functions that fulfill this requirement.) On the global domain $\Omega = \bigcup_i \Omega_i$ all test functions are united to one global test function. Thus, only the boundary terms at the ends of the Purkinje network are left.

Furthermore, equation (4.27) can be used to define a single potential $V$ and formulate the problem as a single variational form:

$$\int_\Omega V_t \, \psi \, dx - \sum_i D \, V_x(E_i) \, \psi(E_i) + D \, V_x(A_0) \, \psi(A_0) + D \int_\Omega V_x \, \psi_x \, dx = 0, \tag{4.31}$$

where $A_0$ is the position of the AV node and the $E$'s are the junctions between the Purkinje tree and the myocardial tissue. The global $D$ is defined from all the local $D_i$'s.

## 4.3 Coupling the Purkinje Network and the Myocardial Tissue

To be able to couple the Purkinje network and the myocardial tissue it is first necessary to know what the equations on each of these domains look like. The equation for the whole Purkinje tree was described in the previous section. For the myocardium a 3D model is necessary. The weak formulation is given by equation (4.21). Except for at the interface between the Purkinje tree and the myocardium, homogenuous Neumann conditions are assumed:

$$\int_{\partial\Omega} (\nabla V \cdot n) \, \psi \, ds = 0. \tag{4.32}$$

This corresponds to the myocardium being electrically insulated from its surroundings, the signal not crossing the borders of the myocardium.

There are many ways to connect the Purkinje network and the myocardial muscle tissue. In Section 2.3, it was briefly described how the exchange of

signals between these two areas happen via transitional cells at the Purkinje-myocardial junctions (PMJs). At the PMJs there is continuity of the transmembrane potential and conservation of current between the Purkinje tree and the transitional cells and between the transitional cells and the myocardium, but the transmembrane potential is discontinuous between the Purkinje network and the myocardium. In this thesis, transitional cells were not explicitly incorporated. Instead, information was exchanged directly between Purkinje cells in the Purkinje network and ventricular myocytes in the ventricular muscle. For all approaches it must be decided which areas should be considered to be part of the interface between the Purkinje network and the myocardium, and how information should be exchanged at these connections. Several approaches were considered and some of them implemented.

### 4.3.1 Coupling Through Boundary Conditions

The Purkinje tree and the myocardium are represented as separate domains. This gives a decoupled problem, which makes it necessary to explicitly assign boundary conditions on both domains to enable two-way signal propagation. To begin with, it was assumed that the terminal point of a Purkinje branch connects to a small area $A_M$ of myocardial surface, typically a single facet. The terminal Purkinje node is denoted by $P$, so $V_P$ is the membrane potential in that node. The area $A_M$ is assumed to contain $k$ myocyte nodes, denoted by $M$, where $M$ is then an integer ranging from 1 to $k$. The membrane potential in such a node is given by $V_M$. To make the notation for the Purkinje and the myocardial problems alike, the end node $P$ in the Purkinje tree is assumed to have cross-sectional area $A_P$. The Purkinje conditions are then formulated in a general dimension, which is in accordance with how it is implemented in FEniCS.

Continuity of voltage now gives:

$$V_P = V_M \tag{4.33}$$

for all nodes $M$ in $A_M$, and conservation of current gives:

$$\int_{A_P} \sigma_P \frac{\partial V_P}{\partial n}\, ds = -\int_{A_M} \sigma_M \frac{\partial V_M}{\partial n}\, ds, \tag{4.34}$$

where the total current out of the Purkinje end equals the total current entering the myocardium at that PMJ area. The variables $\sigma_P$ and $\sigma_M$ [mS/cm] are the conductivities in the Purkinje and the myocardial tissues, respectively.

Two alternative implementations of equations (4.33) and (4.34) were considered. The first alternative is a Neumann condition for the Purkinje node

and Dirichlet conditions for the myocardial nodes:

$$\sigma_P \frac{\partial V_P^{n+1}}{\partial n} = -\frac{1}{A_P} \int_{A_M} \sigma_M \frac{\partial V_M^n}{\partial n}\, ds, \tag{4.35}$$

$$V_M^{n+1} = V_P^n, \tag{4.36}$$

where the superscript denotes the time step.

Next is the reverse: Dirichlet condition for the Purkinje node and Neumann conditions for the myocardial nodes:

$$V_P^{n+1} = \frac{1}{k} \sum_{M=1}^{k} V_M^n, \tag{4.37}$$

$$\sigma_M \frac{\partial V_M^{n+1}}{\partial n} = -\frac{1}{A_M} \int_{A_P} \sigma_P \frac{\partial V_P^n}{\partial n}\, ds. \tag{4.38}$$

These two methods ensure continuity of membrane potential and conservation of current at the PMJs. To be able to include the condition in equations (4.35) and (4.38) as a natural boundary condition in equation (4.21), it must be expressed with a diffusion coefficient $D$ rather than a conduction coefficent $\sigma$, where $D = \sigma/(\chi C_m)$. This was also done when coupling the branches in the Purkinje network. Then $\chi$ and $C_m$ were assumed to have the same values on all branches, so the resulting condition for conservation of current was equal to the original formulation. Here the coupling is between Purkinje cells in the Purkinje network and ventricular myocytes in the myocardium. The cell models used (PRd for Purkinje cells, Decker et al. for ventricular myocytes) use the same value for $C_m$, but since the cells volumes of Purkinje cells and ventricular myocytes differ, $\chi$ is likely to have different values for the two tissue types. Assuming that $\chi_P = \chi_M$, a possible source of error is introduced in the model, but allows conservation of current in equation (4.34) to be expressed as:

$$\int_{A_P} D_P \frac{\partial V_P}{\partial n}\, ds = -\int_{A_M} D_M \frac{\partial V_M}{\partial n}\, ds. \tag{4.39}$$

The corresponding changes in equations (4.35) and (4.38) are substitution of $\sigma_P$ and $\sigma_M$ with $D_P$ and $D_M$, respectively.

A third alternative was also considered. This fulfills equation (4.39), so current is (almost) conserved, but this method allows for a discontinuous membrane potential at the PMJs. This alternative is Robin conditions for both sides. The flux $[\mu\text{A/cm}^2]$ out of the Purkinje tree depends on voltage

differences, and so does the flux into the myocardium:

$$-\sigma_P \frac{\partial V_P^{n+1}}{\partial n} = \frac{\sigma_{PMJ}}{A_P}(V_P^{n+1} - \frac{1}{k}\sum_{M=1}^{k} V_M^n), \tag{4.40}$$

$$\sigma_M \frac{\partial V_M^{n+1}}{\partial n} = \frac{\sigma_{PMJ}}{A_M}(V_P^n - V_M^{n+1}). \tag{4.41}$$

Here $\sigma_{PMJ}$ can be seen as the conductance [mS] (not conductivity, so $\sigma_{PMJ}$ is not directly comparable to $\sigma_P$ and $\sigma_M$) of the Purkinje-myocardium coupling, and can represent the high-resistance barrier of the transitional cells. As before, $\chi$ and $C_m$ are assumed to be the same for all cell types (here also transitional cells), and included on both sides to make the conditions natural boundary conditions in the weak form. Along the line of how it was defined for the Purkinje and myocardial diffusion coefficients, one can define $D_{PMJ} = \sigma_{PMJ}/(\chi C_m)$ [cm$^3$/ms] (which is not directly comparable to $D_P$ and $D_M$ because of different units).

The first and third alternatives were implemented. With some adjustments (which will be further elaborated on in Chapter 5), they were both able to give both anterograde and retrograde propagation. However, none of them were able to reproduce defined delay times for the transmission between the Purkinje network and the myocardial tissue (see Section 2.3).

### 4.3.2 Coupling Through a Stimulus Current

To achieve substantial delay times, the approach of Deo et al. [6] and [7], Boyle et al. [3] and Vigmond and Clements [26] was tested. As for the cases already described, the mentioned articles couple one Purkinje node to all the myocyte nodes in an area $A$. Here $A$ is the area within a specified radius from the Purkinje cell. The current flowing from the Purkinje cell into the myocardium is implemented as a boundary condition when solving for the Purkinje system, and as a stimulus current when solving for the myocardium (Vigmond and Clements [26]). The current [$\mu$A] in a Purkinje branch is given by:

$$i_L = -\pi\rho^2\sigma_i \frac{\partial \phi}{\partial x}, \tag{4.42}$$

where $\rho$ [cm] is the radius of the branch, $\sigma_i$ [mS/cm] is the intracellular conductivity and $\phi_i$ [mV] is the intracellular potential. This equation is quite similar to the left hand side of equation (4.34). At the PMJs, the expression for the current is taken to depend on the voltage differences across

the junction gap:

$$i_L = \sum_k \frac{\phi_i(x_e) - \phi_i^M(x_k)}{R_{PMJ}}, \tag{4.43}$$

where $\phi_i(x_e)$ is the intracellular potential of the terminal Purkinje cell in point $x_e$ and $\phi_i^M(x_k)$ is the intracellular potential of the $k$'th myocardial node inside the junctional radius. The resistance at the PMJs, $R_{PMJ}$ [1/mS], is between one Purkinje node and one myocyte node. Each term in the sum thus gives the electrotonic load of one myocyte node on the Purkinje node, so to get the total loading of the tissue on one end in the Purkinje tree, the sum of all these individual loads must be added. In Deo et al. [6] the last equation is multiplied by a factor $K_{PMJ}$ [dimensionless], which is described as a loading factor to account for sink effects from surrounding tissue that is not directly coupled.

For the boundary condition in the Purkinje problem, this approach is really just the Robin condition above. The difference in formulation is that the Robin condition in equation (4.40) use the average value of the myocyte nodes, while Vigmond and Clements use the sum of individual differences. The two approaches are equivalent if the correct resistances are used. Using the notation introduced earlier in this thesis, equation (4.43) from Vigmond and Clements can be written in the following way:

$$\sum_k \frac{\phi_i(x_e) - \phi_i^M(x_k)}{R_{PMJ}} = \sum_{M=1}^{k} \frac{V_P - V_M}{R_{PMJ}}, \tag{4.44}$$

and then further rewritten:

$$\sum_{M=1}^{k} \frac{V_P - V_M}{R_{PMJ}} = \frac{k}{R_{PMJ}}(V_P - \frac{1}{k}\sum_{M=1}^{k} V_M) = \frac{1}{r_{PMJ}}(V_P - \frac{1}{k}\sum_{M=1}^{k} V_M), \tag{4.45}$$

where $r_{PMJ} = R_{PMJ}/k$ is the resistance between the Purkinje node and the collection of myocyte nodes to which it is coupled. This resistance is a factor $k$ smaller than the individual resistance between two nodes. The last term is on the form used in equation (4.40) for the Robin condition. Equating equations (4.42) and (4.43) for $i_L$ at the PMJs, using the substitution from equation (4.45) for equation (4.43), using $\sigma_P = \sigma_i$, $A_P = \pi\rho^2$ and $1/(r_{PMJ}) = \sigma_{PMJ}$ for conductance, results in equation (4.40).

For the myocardial problem, Vigmond and Clements [26] state that the current given by equation (4.42) is used as stimulus current at the PMJs. Since this is the total current [$\mu$A] and a cell models models the currents per area [$\mu$A/cm$^2$] or per area divided by $C_m$ [$\mu$A/$\mu$F], it is not completely clear

how this current is applied as stimulus current, but if one assumes that it is divided by the myocardial PMJ area the stimulus measured in $\mu$A/$\mu$F can be written:

$$I_{stim} = -\frac{1}{C_m}\frac{A_P}{A_M}\sigma_P\frac{\partial V_P}{\partial x}. \tag{4.46}$$

In this thesis $\sigma$ and $\chi$ are lumped together in the diffusion coefficient $D$, and the individual values for $\sigma$ and $\chi$ are not known. The expression used in the place of equation (4.46) is:

$$I_{stim} = -D_P\frac{\partial V_P}{\partial x}. \tag{4.47}$$

The difference between these two equations is that the dimensionless factor $A_P/A_M$ in equation (4.46) is represented by $1/\chi$ with units cm in equation (4.47). This last term gives equation (4.47) the wrong units but ignoring this, the term $1/\chi$ can be seen as a representative of the relationship between the PMJ areas in the Purkinje tree and the myocardium. With a known value for $\chi$ this solution could have been improved, but as it is it can probably at least partly account for the scaling necessary in the implementation.

Different myocardial PMJ areas were tested. First the nodes of a single facet were used, then the whole cell of that same facet, and then several cells building from this single cell.

# Chapter 5

# Implementation

To model the electrical activity in the ventricles, the monodomain equation with cell models should be solved on the Purkinje network domain and on the myocardial tissue domain on a time interval. In the previous chapter it was explained how an approximate solution at discrete times and discrete spatial points could be found by solving a coupled system of ODEs in each spatial point and a linear PDE, for each time step. A solver must then have one part responsible for solving the system of ODEs, and one part for solving the PDE. This first part will be presented in Section 5.3, and the second in Section 5.4. The equations are defined for the Purkinje network and the myocardial tissue, the meshes representing these domains are described in Section 5.2. The coupling of the two meshes is treated in Section 4.3.

The problem is defined in two separate classes, one for each of the two meshes, and the class for the Purkinje tree holds an object of the class for the myocardial tissue. Coupling information must be sent from the Purkinje tree to the myocardium, and in the myocardium class it must be stored in variables that can be accessed by the Purkinje tree. The class for the Purkinje network controls the solving sequence. It contains a method `solve` that has a loop over all time steps. For each time step a method `step` is called in both solver-classes, that integrates both the system of ODEs and the PDE one time step forward. In addition to these main methods, both classes have many methods and much structure in connection with visualization and the coupling of the meshes.

Beginning with a single cell, the implementation proceeds gradually towards solving the equations on the whole Purkinje network, and then for both the Purkinje network and the myocardial tissue. This process will be described in detail, but first a presentation of the framework.

## 5.1 Programming Packages and Received Code

The code developed during the work with this thesis is written in the programming language Python. Some programming packages that have been important for the work but might not be generally known are presented in this section, as well as code that has been provided by other people, but is not publicly accessible.

### 5.1.1 FEniCS

> "The FEniCS Project is a collaborative project for the development of innovative concepts and tools for automated scientific computing, with a particular focus on automated solution of differential equations by finite element methods."

This is how FEniCS presents itself in its "about" section on its webpage [10]. As stated, FEniCS provides an easy interface to solve many different kinds of differential equations using finite element methods. It consists of different components written in C++, and provides among other things a Python interface. FEniCS is an open source project, and can be downloaded for free. The complete code can be found in a bitbucket repository, the link of which can be found on the FEniCS webpage [10]. One of the main contributors to the FEniCS Project is Simula Research Laboratory.

In this thesis FEniCS is used to define the whole problem, and to solve the PDE part of the problem.

### 5.1.2 Gotran

Gotran stands for "general ODE translator". It is a programming package providing among other things a Python interface to declare arbitrary ODEs. It can also load models from external ODE desciption files such as CellML. The code is developed by Johan Hake and can be found in his Bitbucket repository [13]. In this thesis, gotran was used to declare the cell models, which are systems of ODEs.

Gotan defines the equations in three parts: `states`, `parameters` and `component`. There are several of these, making the file well arranged. For potassium for example, one can define:

```
states("K",
       K_i = ScalarParam(136.422946, unit="mmole*l**-1"))

parameters("K",
        zK = 1.)

component("K")
dK_i_dt = -IKtot*Acap/((Vmyo + Vss_CaL + Vss_sr)*zK*F)
```

The three parts are given the same name, `K`, to group them together. The variable `states` defines the state variables of interest in this group, and gives them initial values, `parameters` should be self-explaining and `component` defines everything else, most importantly the ODEs, but also all other equations and variables included in the cell model. It is not strictly necessary to split the code into these three parts, but for large systems of ODEs it makes it much easier to get an overview of the model.

### 5.1.3 Goss

Goss stands for "general ODE system solver". In this thesis it is used to solve the system of linear ODEs resulting from operator splitting on the monodomain equation for each point in the Purkinje and myocardial grids. The ODEs are specified in gotran format. The code is developed by Johan Hake and can be found in his Bitbucket repository [11].

### 5.1.4 Plotting Packages

FEniCS uses vtk as standard. This gave reasonable visualization for wave simulations in 1D, as can be seen for example in Figure 6.1, and for simulations on the myocardial mesh, see Figure 6.14 (left). However, it failed for the Purkinje tree, visualizing only the topological dimension (1D), as can be seen in Figures 6.8 and 6.9. Using vtk imported in Python, the Purkinje tree could be visualized in 3D by drawing lines between two consecutive points, giving the whole line between two points a color corresponding to the middle value of the potential in the two points. This worked reasonably well for small trees, up to about 40-50 branches, for a short time scale, but a bigger tree and longer time scale was too demanding. The resulting visualization can be seen in Figure 6.10.

The plotting package Mayavi2 was used for plotting the Purkinje tree in 3D, as can be seen for example in Figure 6.11. For efficiency, all the branches in the Purkinje tree were combined to define a single object. This only needs to be done once, so at each time step the only necessary change is to update

the potential value in each point. This plotting is much less demanding than vtk and looks much better for the Purkinje tree used in this thesis. Mayavi2 for the Purkinje tree and FEniCS's vtk for the myocardium were used for real-time visualizations of the electrical activity in the ventricles, see Figure 6.14.

Paraview was used for plotting the Purkinje and myocardial meshes separately and together, see for example Figure 6.16 (d-f). All the information (coordinates and values) is written to file for each time step, and one can read as many meshes into Paraview as desired, and they will be placed correctly with respect to each other. This was the superior package for 3D plotting. It did not offer real-time visualization (which makes it faster), so Mayavi2 and FEniCS's vtk were still useful for testing purposes, but Paraview saves everything, so once a simulation has been run, all time steps can be visited separately and all viewing settings can be chosen as desired. The last fact means for example that the mesh can be rotated so as to be seen from all angles as desired, it is possible to change time and color scales, and to make cuts in the mesh. Isochrone maps are also shown in Paraview. For a cut isochrone map, see for example Figure 6.16f. The only drawback with saving each time step in Paraview is the large file size.

Python's pylab was used for time series in points, see for example Figure 6.16a, and for comparison between graphs in 1D, as seen for example in Figure 6.2.

### 5.1.5 PRd Cell Model and Monodomain Solver

Code implementing the PRd cell model in the C programming language can be found through a link [4] that is given in the original publication of the model [17]. For this thesis it was provided by Pan Li. The code for the PRd model used in this thesis is based on this code.

The code for solving the monodomain equation is based on existing code developed at Simula Research Laboratory. The original code provided a simple, general 3D solver.

### 5.1.6 Code to Define the Myocardium and Generate a Purkinje Network

Code to generate a Purkinje network was provided by Glenn Terje Lines and Siri Kallhovd. Lines's code can be used to generate and visualize a realistic Purkinje network in the ventricles. Kallhovd's code modifies and adds to Lines's code. The following files were used (and modified as seen fit), all of which are from Kallhovd:

- `diastole115_purkinje.msh`: This is a text file with numbers that define the myocardial mesh. The information is given as nodes and elements. From MR images from a human patient, 3584 points spread out on the whole myocardial tissue give the node coordinates. An element is specified by listing which nodes it is drawn up between. There are several types of elements. Line elements define the outer edge of the base of the ventricles. Triangle elements are separated into four groups: the endocardial surface of the left and right ventricles (two groups), the epicardial surface and the surface of the base. Tetrahedra elements define the volume of the ventricles.

- `call_tree_msh.py`: This file reads the necessary information about nodes and elements from `diastole115_purkinje.msh`. From this it finds a node in the base plane that is suitable as a common root node for the Purkinje trees in the left and right ventricles. The trees themselves are built by file `get_tree_msh.py`. At last a method in the file `vtk_transform_purkinje.py` refines the network. The information about the network is stored in a dictionary called `cable_dict`.

- `get_tree_msh.py`: Given a root node and an endocardial surface (elements), this file builds the tree from base (where the root is) to apex. Starting at the root node, all nodes in all elements containing this node are evaluated. If a node among those being evaluated is not already in the tree and is more apical than the root node, this new node is added to the tree. For each node that was added during this "level", this process is repeated, so the tree is built generation-wise until it reaches the apex. The original code allows a junction to split a branch into three new branches, this was modified to only allow two new branches. The information about the tree is stored in several variables, among them the dictionary `cable_dict`.

- `vtk_transform_purkinje.py`: This file has several methods, but only the method `tree_to_carp_purkinje_format` is used in this thesis. Given a Purkinje tree (in the form of `cable_dict`) and a resolution `ds` (given in cm), it refines the network so that each branch stores the coordinates of points along it with approximate distance `ds` between them. These points are called local nodes and have local numbering on each branch.

From the above descriptions it should be clear that the dictionary `cable_dict` is important. Information about a specific cable or branch can be reached through `cable_dict["cable_nr"]`. Assigning this to a new variable `cable`,

the number of local nodes it is made up of is given by `cable["nr_nodes"]`, and the coordinates of local node $i$ can be found in `cable["Node_i"]`, the node numbers increasing from less to more apical. The cables are numbered generationwise (this follows from the process described for `get_tree_msh.py`), first the right ventricle, then the left (this is just the order of building the trees in `call_tree_msh.py`).

Code for visualizing the Purkinje tree in matplotlib could be found in the file `plot_tree_msh.py`. This was only used for initial inspection of the tree.

## 5.2 Meshes

As described in Chapter 2, the Purkinje network is a network of specialized cells spreading from the SA node out on the endocardial surface of the ventricles, the cells on the ends of the network being (indirectly via transitional cells) connected to cells in the myocardial tissue, allowing signals to pass between the Purkinje network and the myocardium. The myocardium is cardiac tissue making up most of the heart wall. Both of these objects must be defined as FEniCS meshes in the implementation.

Some simpler meshes are used as test cases. These are most often unit intervals, squares and cubes, and are created very easily in FEniCS:

```
1Dmesh = UnitIntervalMesh(nx)
2Dmesh = UnitSquareMesh(nx, ny)
3Dmesh = UnitCubeMesh(nx, ny, nz)
```

Here `nx` signifies the number of elements in the $x$-direction, and correspondingly for `ny` and `nz`. Other types of meshes will be presented where they occur.

### 5.2.1 The Purkinje Network

The Purkinje network is modeled as a collection of branches in 1D. More specifically each branch is modeled as many short 1D elements in FEniCS, i.e. they are intervals, but all vertices have 3D coordinates. This gives a 3D tree, but built with the simplest elements possible, creating much less data than if a complete 3D environment was to be used. Combining a 1D and a 3D framework in this way is achieved through MeshEditor, which is a feature in FEniCS that can be used to build meshes with different topological and geometrical dimensions. The Purkinje tree has topological dimension one (1D elements) and geometrical dimension three (3D coordinates). The 389 branches in the tree are built up of a total of 6357 nodes and 6356 elements.

The information about the Purkinje tree comes from the code from Lines and Kallhovd. The tree is completely defined by the dictionary `cable_dict` and the array `N`, where all the node coordinates are stored, but not on a form that allows directly building of a FEniCS mesh with MeshEditor. To build the mesh one must have a global numbering for all the nodes and elements, and for each element one must know which nodes it is built up of.

First the total number of elements and vertices must be known. The number of vertices is found by adding all local nodes on all the cables, making sure no node is counted twice. The number of elements is one less than the number of vertices. The code for this is:

```
n_cables = len(cable_dict)
nr_nodes = 1
for i in xrange(n_cables):
    nr_nodes += cable_dict[str(i)]["nr_nodes"] - 1
```

If a smaller tree is wanted, the number of cables can be set to any number in the interval $[3, 839]$. Three is the lowest number because this corresponds to the first three branches connecting the Purkinje trees in the left and right ventricles, while 839 is the length of `cable_dict`. Next a mesh and a MeshEditor are initialized:

```
mesh3d = Mesh()
e = MeshEditor()
e.open(mesh3d, 1, 3)
```

The last line gives 1 as topological dimension (the dimension of the elements) and 3 as geometric dimension (the dimension of the vertices). Then the vertices are initialized, created and added, and the same for the elements:

```
e.init_vertices(nr_nodes)
e, firstNodes_dict = create_nodes(cable_dict, e, n_cables)

e.init_cells(nr_nodes - 1)
e = create_elements(cable_dict, e, firstNodes_dict, n_cables)
```

The actual creation of vertices and elements are done in separate methods. The main part of method `create_nodes` is a loop over all the cables, calling method `create_nodes_sub`, which again loops over all the local nodes on the given cable, adding all but the first node as vertices. The exception of the first node is to avoid duplicates, because the first node on a cable is also the last node on another cable for all cables except the first. The first node on the first cable is added separately in the beginning of method `create_nodes`. The global node number increases with one for each added vertex. To add

a vertex one needs to know the global node number and the position. The rest of the code in these two methods is for use when creating the elements.

```
def create_nodes(cable_dict, e, n_cables):
    pos = cable_dict["0"]["Node_0"]    #first node on first cable
    e.add_vertex(0, pos[0], pos[1], pos[2])
    firstNode = str(pos[0]) +  str(pos[1]) + str(pos[2])
    #first node on every cable.
    #Position is key, global node number is value
    firstNodes_dict = {}
    firstNodes_dict[firstNode] = 0
    start = 1

    for i in xrange(n_cables):
        e, start = create_nodes_sub(cable_dict, e, firstNodes_dict, \
i, start)
    return e, firstNodes_dict


def create_nodes_sub(cable_dict, e, firstNodes_dict, cable, start):
    cable = cable_dict[str(cable)]
    cable["secondNode"] = start  #global node number of Node_1
    nr_nodes = cable["nr_nodes"]
    for i in xrange(nr_nodes-1):
        pos = cable["Node_"+str(i+1)]
        e.add_vertex(start+i, pos[0], pos[1], pos[2])

    pos = cable["Node_"+str(nr_nodes-1)]
    #might be the first node on another branch
    endNode =  str(pos[0]) +  str(pos[1]) + str(pos[2])
    firstNodes_dict[endNode] = start + nr_nodes - 2
    return e, start + nr_nodes - 1
```

The main parts of the methods for creating the elements are the same as for the vertices: a loop over each branch, and for each branch a loop over all local elements, but there is one extra consideration to make. To add an element one has to know the global element number and the global node number of the vertices in the element.

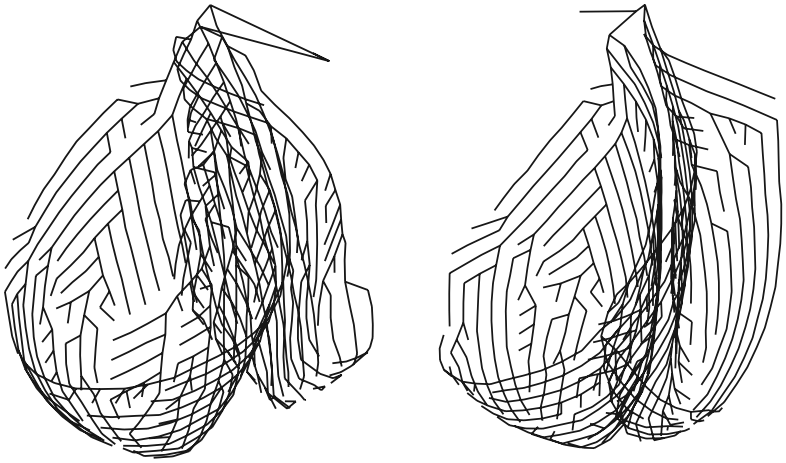
The global element number increases with one per added element, as for the nodes, therefore the global element number of the next element to be added is always known. However, for the first element on a random branch, the global node numbers of the two first nodes on the branch (which define the element) are not known. If the global number of the second node is known, the global number of the more apical local nodes are known, as they simply increase by one per node, but for the first two nodes on each branch, special measured must be made.

The solution here is to add the coordinates and global node number as a key-value pair in a dictionary `firstNodes_dict` for the first node on each branch when the nodes are created, while the global number of the second node is stored as a separate entry in `cable_dict`. When creating an element the node coordinates are known from `cable_dict`, so the global node number of the first local node can be found from `firstNodes_dict`, while the global node number of the second local node can be found directly in `cable_dict`. The reason why the global numbers of the first and second nodes are stored differently is that when creating global nodes on a branch, the global number of the second node is always known because it is created on that branch, while the first node was created on an earlier branch, so the global node number is not known, only the coordinate.

```
def create_elements(cable_dict, e, firstNodes_dict, n_cables):
    start = 0
    for i in xrange(n_cables):
        e, start = create_elements_sub(cable_dict, e, firstNodes_dict, \
i, start)
    return e


def create_elements_sub(cable_dict, e, firstNodes_dict, cable, start):
    cable = cable_dict[str(cable)]
    nr_nodes = cable["nr_nodes"]
    pos = cable["Node_0"]
    firstNode =  str(pos[0]) +  str(pos[1]) + str(pos[2])
    firstNode = firstNodes_dict[firstNode]
    secondNode = cable["secondNode"]
    e.add_cell(start, firstNode, secondNode)
    currNode = secondNode
    for i in xrange(1, nr_nodes-1):
        e.add_cell(start+i, currNode+i-1, currNode+i)
    return e, start + nr_nodes - 1
```

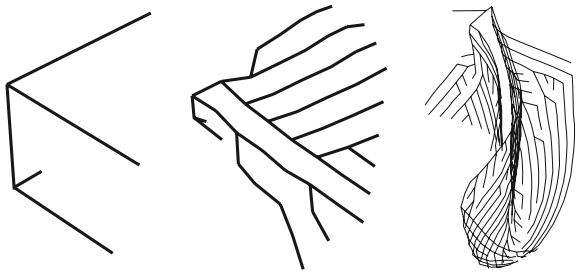
The resulting mesh is shown in Figure 5.1a. It is not completely identical to the original tree, because that tree allowed for junctions to split a branch in three, while this tree only allows splitting into two new branches. This is controlled in file `get_tree_msh.py`.

In the original mesh shown in Figure 5.1a, the connection between the ventricles has very long cables compared to the cables in the two trees. This was adjusted in the file `call_tree_msh.py`. The code for this is quite easy, but requires some tedious adjustments of `cable_dict` entries, which is not shown here. The tree where the first three branches are adjusted is shown in Figure 5.1b.



**(a)** Original **(b)** Modified start

**Figure 5.1:** Purkinje tree mesh



**(a)** 5 cables **(b)** 50 cables **(c)** 500 cables

**Figure 5.2:** Different sizes of Purkinje tree mesh

Some trees for other values of `n_cables` are shown in Figure 5.2. One can see that the tree for the right ventricle is built first.

### 5.2.2 The Myocardial Tissue

The myocardial tissue is modeled in 3D by tetrahedra. The mesh consists of 3584 vertices and 10 931 elements and can be seen in Figure 5.3. The grid is quite coarse, and it is possible to see some of the individual elements.

Meshes in FEniCS are often defined as xml-files that can be read directly into FEniCS. This mesh is defined by nodes and elements that can be found in the file `diastole115_purkinje.msh`. This is practical with respect to the Purkinje tree since the nodes of the tree can then be made from nodes on the myocardial surface, but it enforces manual creation of the nodes and

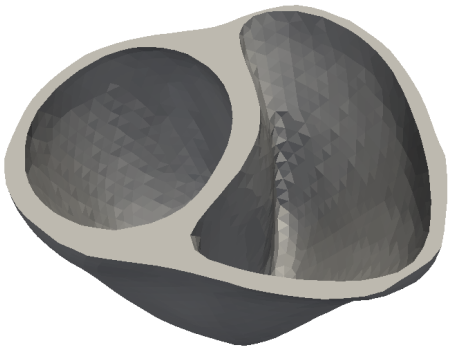


**Figure 5.3:** Mesh for the myocardial tissue.

elements. This is done using MeshEditor, as for the Purkinje mesh, but it is much easier to make than the Purkinje mesh. A mesh and a MeshEditor is created as before, but with algebraic and geometric dimensions both equal to 3. The node information of interest is read into `N`, and the element information into `E`, and all that is left is a simple looping:

```
e.init_vertices(len(N))
for i in xrange(len(N)):
    e.add_vertex(i, N[i,0], N[i,1], N[i,2])

e.init_cells(len(E))
for i in xrange(len(E)):
    e.add_cell(i, E[i,0], E[i,1], E[i,2], E[i,3])
```

The nodes in the original Purkinje network (before the refinement resulting from `vtk_transform_purkinje.py`) are also nodes in the myocardial mesh, but after refinement of the Purkinje tree these nodes no longer have the same numbering in the Purkinje and myocardial meshes.

## 5.3 Cell Models

In the previous chapter it was explained how operator splitting on the monodomain equation resulted in a linear system of ODEs and a linear PDE that could be solved sequentially. The system of ODEs is defined by a cell model, in this thesis the PRd model is used for Purkinje cells and the Decker et al. model is used for ventricular myocytes. Here it is shown how this system is solved numerically.

A cell model can be included in the program by direct implementation of the equations in the model, and stepping all of the ODEs one time step

forward with a numerical method. This was done for test cases, using the midpoint method to step the solution forward. For large systems of ODEs this is an impractical approach since the midpoint method must be implemented for each ODE. Luckily goss has a FEniCS interface that can be used. All that is needed is to define the cell model in a speficied format, and to define an appropriate goss-solver. When the cell model is used in space, it is used together with a tissue model, in this case the monodomain equation.

### 5.3.1 A Single Cell

As described above, two different cell models were used in this work. The Decker et al. cell model [15] can be found in CellML format on CellML's website [5]. Running `cellml2gotran` in gotran, with the CellML file as command line argument, this was translated to the file `decker_2009.ode` in gotran format. Thereafter the membrane potential, called `Vm`, had to be renamed `V`, so that the model could be run with gotranrun.

Next, the PRd model was to be defined in gotran format. This file was written from scratch, reusing parts of the code from the `decker_2009.ode` file. To begin with, equations for the intracellular calcium concentrations without buffer adjustments were used, and the stimulus current was given as:

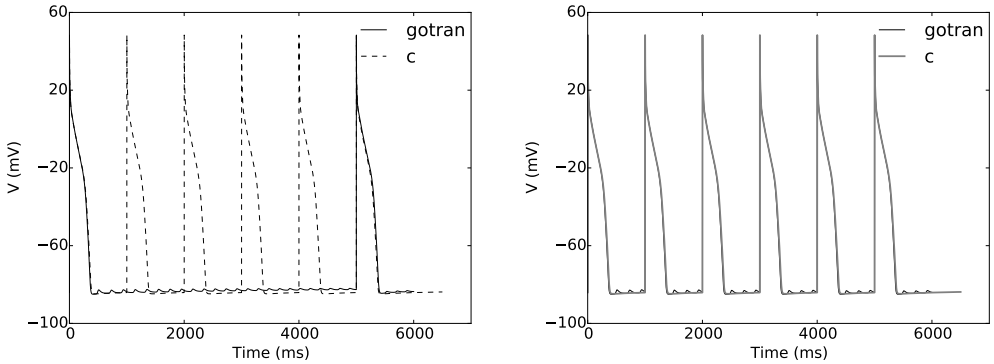
```
I_stim = Conditional(And(Ge(time - past, stim_offset), Le(time - past,
    stim_offset + stim_duration)), stim_amplitude, 0)
```

The resulting plot can be seen in Figure 5.4a. One can see that the solution given by gotran have too few peaks compared to the solution given by the original C program, and it has small oscillations in the resting potential that should not be there. Too few peaks is due to the goss-solver missing out on some of the very abrupt upstrokes. This can be remedied by using a different expression for the stimulus current. If the following expression for stimulus current is used instead, the model gives the right number of peaks, see Figure 5.4b:

```
I_stim = stim_amplitude*ContinuousConditional(Ge(time - past,
    stim_start), 1, 0, 0.2)*ContinuousConditional(Le(time - past,
    stim_start + stim_duration), 1, 0, 0.2)
```

Varying the parameter that here is set to 0.2 will produce various number of peaks, and will also affect an offset in the beginning that can be seen in plots showing only a short timespan, as in Figure 5.5e.

The PRd cell model includes calcium buffers in all the compartments except the NSR compartment. The formulation for the PCS compartment is



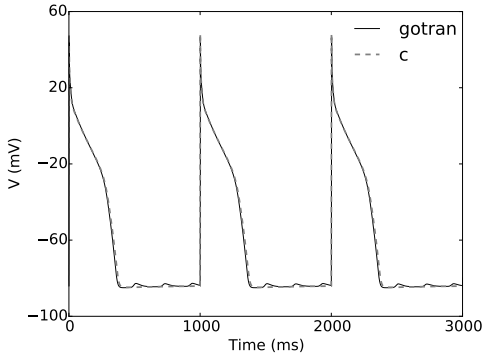

**(a)** Conditional stimulus **(b)** ContinuousConditional stimulus

**Figure 5.4:** Action potentials of a Purkinje cell for the PRd cell model for C and gotran code. In (a) the stimulus function is the same as in the gotran version of the Decker et al. cell model, while in (b) a more complicated stimulus function is used.

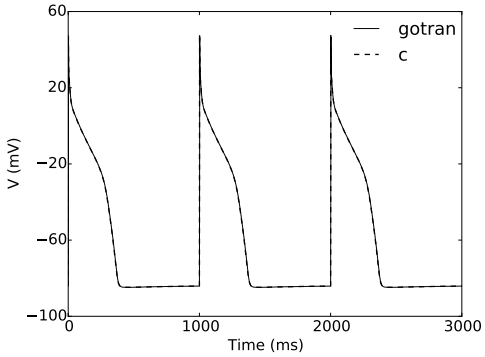
on ODE form, while the others are formulated as solutions of second or third degree polynomial equations, as described in Section 3.1.3. To begin with, buffers were not included in the implementation of the PRd cell model for those formulations that were not on ODE form because they could not easily be defined on gotran form. The resulting action potential had oscillations in the resting phase, as can be seen in Figure 5.5a. All buffer equations were then implemented on ODE form in gotran. When this was done, the oscillations almost disappeared, see Figure 5.5b. Details of the oscillations can be seen in Figures 5.5c and 5.5d for formulations excluding and including calcium buffers, respectively. This demonstrates how the calcium buffers contribute in stabilizing the membrane potential.

In Figure 5.5d one can also see how the Purkinje potential slowly depolarizes during phase 4 of the action potential. This is caused by the funny current $I_f$, as described in Section 2.4.3.

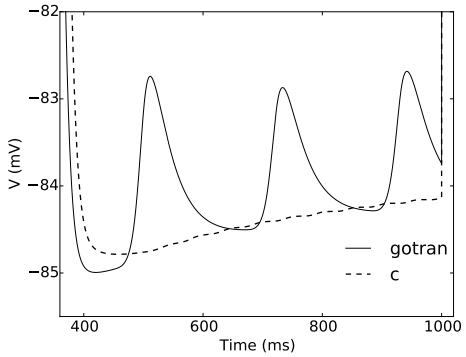
Looking at the plot with the buffer adjusted equations at a shorter time scale, reveals a small offset in the beginning compared to the plot from the C program, see Figure 5.5e, but the offset is so short in time this is not a problem.



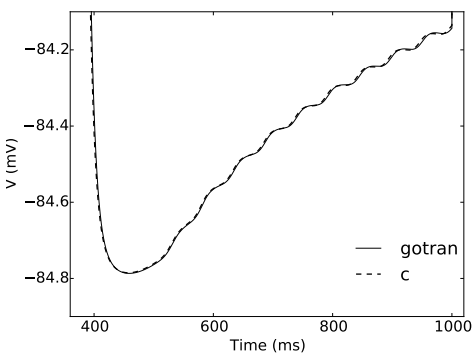

**(a)** PRd



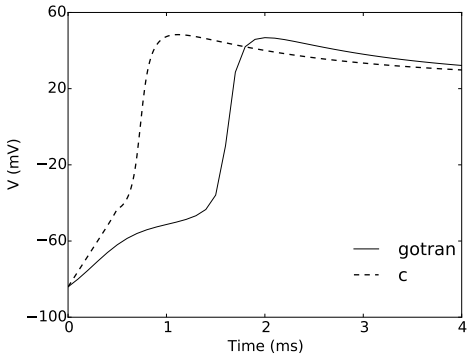

**(b)** PRd




**(c)** detail from (a)




**(d)** detail from (b)




**(e)** detail from (b)

**Figure 5.5:** Action potentials of a Purkinje cell for the PRd cell model for C and gotran code. In (a) most of the buffer effects are left out, while in (b) they are included on ODE form. (c) is a detail from (a), (d) and (e) are details from (b). (a) is a cropped version of Figure 5.4b

### 5.3.2 Cell Models in Space: Goss-Solver

With a coupled system of ODEs in gotran format, that part of the problem can be solved by a goss-solver, which steps the ODEs one time step forward:

```
goss_solver.step((t, t+dt), self.v_p)
```

Here `self.v_p` is the solution at the previous time step, i.e. $V^n$ if the problem involves $V^n$ and $V^{n+1}$. To create a goss-solver, one or more cell models in gotran format are loaded into the program, and assigned to the correct areas. Together these areas make up the mesh where the problem is defined. If there is one cell model on the whole mesh, this can be coded as:

```
def ode_system(self):
     """Creates a goss ODESystemSolver"""
     cellmodel = goss.dolfin_jit(load_ode("fhn.ode"),
     field_states = ["v"])
     solver = goss.DOLFINODESystemSolver(self.mesh(), cellmodel)
     return solver
```

Here the cell model defined in the file `fhn.ode` is used for the whole mesh. If there are several cell models, it must be specified which models are to be used where. In this case, different cell models come in the form of one part of the domain receiving a stimulus, while the rest of the domain does not. Since these two cell models differ by a stimulus term, they count as different models. The method `goss.DOLFINODESystemSolver` now takes a MeshFunction as an additional argument, marking the different areas of the domain, and the `cellmodel` argument must be a dictionary mapping the correct cell model to the correct area. The new code can be written as:

```
class StimDomain(SubDomain):
    def inside(self, x, on_boundary):
        something...

# Create mesh functions over the cell facets
domains = MeshFunction("size_t", mesh, mesh.topology().dim() - 1)
domains.set_all(0)
mesh_stim = StimDomain()
mesh_stim.mark(domains, 1)

cellmodels = {1:cellmodel_stim, 0:cellmodel}
solver = goss.DOLFINODESystemSolver(mesh, cellmodels, domains)
```

The goss-solver contains a loop in C that loops over all the nodes in the mesh, solving the specified system of ODEs for each node over the given time interval with a Newton solver.

## 5.4 Tissue Model

In Section 4.1.1 it was shown how the solution of a coupled system of ODEs coupled to a PDE could be approximated by splitting the problem into a system of ODEs and a linear PDE and solving these two parts sequentially. The implementation of the system of ODEs was explained in the previous section. This section will explain the implementation of the linear PDE. This is used for both the Purkinje tree and the myocardial tissue. The explanation will focus on the Purkinje fibres, but the implementation for the myocardium is identical. The tissue model used in this thesis is the monodomain equation.

### 5.4.1 The Linear PDE

In Section 4.2.2, it was shown that the linear PDE resulting from operator splitting on the monodomain equation could be written as a single weak form on the whole Purkinje domain:

$$\int_\Omega V_t\,\psi\,dx - \sum_i D\,V_x(E_i)\,\psi(E_i) + D\,V_x(A_0)\,\psi(A_0) + D\int_\Omega V_x\,\psi_x\,dx = 0. \quad (5.1)$$

Assuming homogenuous Neumann conditions on all the boundaries for now, the boundary terms vanish. Using the specified time discretization and a more dimensionally general form for the derivative in space gives:

$$\int_\Omega (V^{n+1} - V^n)\,\psi\,dx + \Delta t\,D\int_\Omega \nabla V\cdot\nabla\psi\,dx = 0. \quad (5.2)$$

In FEniCS this is implemented as:

```
form = ((v-v_p)*psi + dt*D*inner(nabla_grad(v), nabla_grad(psi)))*dx
```

It should be clear what variables in the code represents what as the FEniCS notation is very similar to the mathematical formulation. The functions are defined like this:

```
v = TrialFunction(scalar)
psi = TestFunction(scalar)
v_n = Function(scalar)
v_p = project(Constant(-84.058830), scalar)
```

where `v_n` is used to store the new solution, and the function space `scalar` is of the continuous Galerkin family and uses first order polynomials:

```
scalar = FunctionSpace(mesh, "CG", 1)
```

It is usual to collect the terms with unknowns on the left hand side and call them $a$, and the terms with only known quantities on the right hand side and call them $L$, such that $a = L$. In the code one can simply write:

```
(self.a, self.L) = system(form)
```

The solution of the weak form is then found by a call to `solve`, which must be repeated for each time step:

```
solve(a == L, self.v_n)
self.v_p.assign(self.v_n)
```

## 5.5 Coupling the Purkinje Network and the Myocardial Tissue

To connect the Purkinje network to the surrounding myocardial tissue of the ventricles, it was assumed that the only interaction happens at the ends of the network, at the PMJs. This is in accordance with physiological knowledge as the Purkinje network is assumed to be electrically insulated from the myocardium at all other places. During normal activation the signal in the ventricles starts at the bundle of His (see overview of the heart in Figure 2.1) at the beginning of the Purkinje network, spreads through all the branches of the tree and travels from the ends of the tree into the myocardial tissue, where it continues until the whole ventricles are activated and the muscles contract. During abnormal heart conditions, there may be retrograde activity as well. For the model to be able to simulate dysfunctional propagation in the ventricles in the best way possible, it is important that the signal is able to move both ways between the Purkinje tree and the myocardial tissue. Some ways to couple these two systems were described in Chapter 4.

### 5.5.1 One-Way Coupling

To begin with, the coupling between the Purkinje network and the myocardial tissue was modeled by homogenuous Neumann conditions for the Purkinje tree (natural boundary conditions) and Dirichlet boundary conditions for the myocardial problem at the connection points, where the Dirichlet values were given by the values at the end points of the Purkinje network. Homogenuous Neumann conditions for the Purkinje tree corresponds to this system being electrically insulated, so no signal can go from the myocardium to the Purkinje tree, while Dirichlet conditions (almost, because of operator splitting)

ensure continuity of the membrane potential at the PMJs, so a signal at the ends in the Purkinje tree will travel into the myocardium. These boundary conditions therefore describe a one-way connection between the Purkinje tree and the myocardium. Results for a myocardial test mesh, and for realistic Purkinje and myocardial meshes can be found in Section 6.2.1.

To be able to apply Dirichlet conditions on the PMJ nodes in the myocardium with values from the PMJ nodes in the Purkinje tree, these Purkinje values must be known, as well as the area to enforce them in the myocardium.

The file `call_tree_msh.py`, which was described in Section 5.1.6, creates an overview over the end nodes in the Purkinje tree. This is an array with length equal to the number of vertices before refinement of the branches, where a "1" signifies that the vertex in that index is an end in the tree, while a "0" means that it is not. From this one can get a set of indices for the ends holding for the unrefined mesh. The coordinates of the ends can be found from looking up these indices in the list `N` of the original nodes. The end values are then found simply by calling the wanted function with the wanted coordinates as input argument.

Dirichlet conditions were then tried enforced by using FEniCS's feature for this, but this procedure proved too inefficient to be of any use. The procedure used could probably have been improved, but there can be no more efficient way than setting the relevant entries in the myocardial function directly, so this approach was chosen. The indices of the PMJ nodes in the myocardial mesh must then be known. All end nodes in the tree mesh are also nodes in the myocardial mesh. Since the myocardial mesh is not changed in any way from the information stored in the file `diastole115_purkinje.msh`, the indices found in `call_tree_msh.py` apply directly for this mesh.

When working directly with FEniCS vectors, one must make sure that the internal numbering in FEniCS is used. This is ensured by using the mapping returned by function `vertex_to_dof_map` on the node numbers given to the nodes when creating them. The following code snippet should clarify the use:

```
vd = vertex_to_dof_map(scalar)
internal_indices = vd[original_indices]
v_p.vector()[internal_indices] = Purkinje_values
```

Here the last line implements equation (4.36), which is continuity of the transmembrane potential $V$.

### 5.5.2 Two-Way Coupling

To achieve a two-way coupling at the PMJs such that the electrical signal can move both from the Purkinje tree to the myocardium and the other way around, some sort of boundary conditions able to transfer signal must be implemented for both the Purkinje problem and the myocardial problem. Four suggestions for such boundary conditions were given in Chapter 4: Neumann conditions for the Purkinje tree and Dirichlet conditions for the myocardium, the other way around, Robin conditions for both, and Robin conditions for the tree and stimulus currents for the myocardium. Of these suggestions, the first, third and fourth were implemented for test cases.

Robin conditions for the tree and stimulus currents for the myocardium were also implemented for the realistic meshes. The results from the test case and the realistic case can be found in Section 6.2.2.

#### Neumann and Dirichlet Boundary Conditions

First (non-homogenuous) Neumann conditions for the Purkinje tree and Dirichlet conditions for the myocardium were implemented. The Dirichlet conditions correspond to continuity of the transmembrane potential, as explained for the one-way coupling, while the Neumann conditions correspond to conservation of current.

The code for these boundary conditions is an extension of the code for one-way connection, which has Dirichlet conditions for the myocardium in one myocardial node per PMJ. Here this is extended to Dirichlet conditions on all three nodes of a facet. This way, the flux through such a facet can be used for the corresponding Neumann condition in the Purkinje tree. The Neumann condition is incorporated by adding the term:

$$- D \int_{\partial\Omega} (\nabla V \cdot n)\, \psi \, ds \tag{5.3}$$

to the weak form, as in equation (4.21), implementation shown in Section 5.4.1. This was done by substituting $D\,\partial V/\partial n$ as given by equation (4.35) for conservation of current, but adjusted according to equation (4.39), giving:

$$D_P \frac{\partial V_P^{n+1}}{\partial n} = -\frac{1}{A_P} \int_{A_M} D_M \frac{\partial V_M^n}{\partial n}\, ds, \tag{5.4}$$

The code for a general number of PMJs, `num_PMJs`, is shown:

```
for i in xrange(num_PMJs):
    form += dt*D_M/FacetArea(mesh)*M_grad[i]*psi*self.ds(i+1)
```

Here `D_M` is the myocardial diffusion coefficient, `M_grad[i]` holds the potential gradient integrated over the $i$'th myocardial PMJ facet, `self.ds(i+1)` (the complete mesh is marked as 0 in the beginning, so the PMJ facets are marked from 1) marks the corresponding facet in the Purkinje tree (which is an end node) and `FacetArea(mesh)` (where `mesh` is the Purkinje mesh) will give the facet area corresponding to `self.ds(i+1)` when integrated over this area. The sign was changed to fit the normal directions in FEniCS.

Since the potential gradients are integrated over the myocardial PMJ facets, it is desirable for efficiency to gather all the integrands in a common expression before calling `assemble`, which performs the integration. In FEniCS this is achieved by multiplying with a test function, in the code denoted `v_grad` from a specialized space `R`, where `v_grad[i]` is then the $i$th unity vector on an appropriate format. As in the code above, `self.ds` marks the facets but now in the myocardium. The definitions of the necessary variables are like this:

```
R = VectorFunctionSpace(mesh, "Real", 0, dim=num_PMJs)
v_grad = TestFunction(R)
gradients = zeros(num_PMJs)
self.ds = ds[facet_domains]
```

Here `facet_domains` is a FacetFunction marking the needed facets with different numbers.

Now that everything that is needed is properly defined, the integrated gradients can be found. This must be done in each time step by running the following code:

```
grad_vals = 0
for i in xrange(num_PMJs):
    grad_vals += Dn(v_p)*v_grad[i]*self.ds(i+1)
grad_vals = assemble(grad_vals)
gradients[:] = grad_vals[:]
```

Initially this code only gave retrograde propagation. To investigate if any problems arose as a result of the connection being between a 1D and a 3D mesh, a unit interval and a unit square were also tested in the place of the unit cube. The 1D-1D coupling gave propagation both ways, while the 1D-2D coupling did not, but the signal travelling from the tree to the tissue was not subdued as fast as in the case of 1D-3D coupling. With some adjustments the model was able to give propagation both ways for both 1D-2D and 1D-3D coupling. Scaling the Neumann condition in the Purkinje problem slightly down (a bit more for 3D than for 2D) made sure the values at the ends of the Purkinje tree were not pulled down so fast by the myocardium, leading

to a longer stimulus (through Dirichlet conditions) of the myocardial nodes.

A drawback with this coupling is that it is under no circumstances able to reproduce any noticeable delay times at the PMJs. This is no surprise, as Dirichlet conditions on the myocardium (almost) ensures continuity of the membrane potential across the junctions.

### Robin Boundary Conditions

Robin boundary conditions were implemented both for the Purkinje and the myocardial problems. This corresponds to conservation of current but allows for disconyinuity of the membrane potential at the PMJs. This type of boundary condition is implemented in a similar way as the Neumann conditions above, acting as natural boundary conditions, but the addition to the weak form looks a bit different. Instead of equation (5.4) above, equation (4.40) adjusted according to equation (4.39) gives:

$$D_P \frac{\partial V_P^{n+1}}{\partial n} = -\frac{D_{PMJ}}{A_P}(V_P^{n+1} - \frac{1}{k}\sum_{M=1}^{k} V_M^n) \tag{5.5}$$

Here the condition for the Purkinje tree is shown. The corresponding condition for the myocardium has replaced the average value of the myocyte nodes with the single Purkinje value. The implementation is as follows:

```
for i in xrange(num_PMJs):
    form += dt*diff_PMJ/FacetArea(mesh)*(v - v_M[i])*psi*self.ds(i+1)
```

where `v_M[i]` is the average value of the myocyte nodes at PMJ area number $i$. This average is not very difficult to implement, but it requires a little bit of extra effort, which will not be shown here. The rest of the variables should be clear from earlier explanations. As before, the sign was fitted to be correct in the FEniCS implementation.

The variable `diff_PMJ` corresponds to $D_{PMJ} = \sigma_{PMJ}/(\chi C_m)$, where $\sigma_{PMJ}$ is the conductance between the Purkinje cells and the transitional cells and between the transitional cells and the ventricular myocytes. Conductance is proportional to conductivity, but they are not the same. The conductivity at the PMJs should have values between the conductivities in the Purkinje tissue and the myocardial tissue. If the PMJ conductivity is too high, the electrotonic load from the tissue on the Purkinje tree through the PMJs will be too high, so the ends of the Purkinje tree will be drained before they have time to activate the myocardium. If the value is too low, the stimulus from the tree to the myocardium is not given enough time to build up before it diffuses into the surrounding tissue in the myocardium. The

variable $D_{PMJ}$ is proportional to the PMJ diffusivity, but they are not the same, so one cannot compare the values of the diffusivities $D_P$ and $D_M$ for the Purkinje tree and the myocardium, respectively, directly with the value for $D_{PMJ}$ at the PMJs.

This code also needed some adjustment for the propagation to work both ways. The Robin condition in the Purkinje problem was scaled up a bit (multiplied by 3), while the Robin condition in the myocardial problem was scaled down a bit (multiplied by 0.3).

### Robin Boundary Conditions and Stimulus Currents

The third and last type of boundary conditions implemented was Robin conditions for the ends of the Purkinje network and stimulus currents for the PMJ nodes in the myocardium. This should ensure conservation of current but allow for a discontinuous membrane potential at the PMJs. The code for the Robin conditions is as above.

To be able to change the stimulus current in the Decker et al. cell model, `I_stim` must be defined as a parameter both in the gotran file and when loading this file into the program. The last part is demonstrated in the code below. This single line is the only change necessary in method `ode_system`, presented in Section 5.3.2:

```
cellmodel = goss.dolfin_jit(load_ode("decker_2009_Istim.ode"),
    field_states=["V"], field_parameters=["I_stim"])
```

The stimulus current can then be set outside the gotran file at any time:

```
self.goss_solver._ode_system_solvers[0].set_field_parameters(I_stim)
```

Here `self.goss_solver` is the solver returned by method `ode_system`, and the 0 signifies subdomain 0. In this thesis the only reason for having several subdomains is to have different stimuli on different areas. When the stimulus current is controlled from the outside there is no reason for splitting the domain, and the whole domain is subdomain 0.

The stimulus variable `I_stim` is a numpy array of the same length as the number of vertices in the mesh, thus the current given to each node is specified. To do so it is necessary to know which entries to give which values. The nodes that should receive a stimulus are those coupled to the ends of the Purkinje tree. Since the myocytes at different PMJs should receive different stimuli, one must keep track of which nodes belong to which PMJs. This information is stored in a numpy array `stim_nodes`, where each entry is a list of the node numbers corresponding to the PMJ given by the array

index. Using the function `vertex_to_dof_map` introduced in Section 5.5.1, the entries of `I_stim` can be filled.

In Section 4.3.2, the stimulus current was defined by equation (4.47) as:

$$I_{stim} = -D_P \frac{\partial V_P}{\partial x}. \tag{5.6}$$

The implementation is given by:

```
I_stim = np.array([0]*mesh.num_vertices(), dtype=np.float)
for i in xrange(len(stim_nodes_dof)):
    I_stim[stim_nodes_dof[i]] += diff_P*P_grad[i]
```

Here `P_grad` is defined almost as `M_grad` (which was not really a gradient) in the implementation of Neumann conditions above. The difference is that now `grad_vals` is divided by the facet area when it is multiplied by `self.ds(i+1)`.

To get propagation both ways this model too needed some adjustment. The Robin condition was scaled up by a factor labeled $K_1$ [dimensionless], while the stimulus current was scaled up by a factor labeled $K_2$ [dimensionless]. Together with the PMJ "diffusion coefficient" $D_{PMJ}$ (described in Section 4.3.1) and the number of myocyte nodes receiving a stimulus, these two factors were balanced against each other to give bidirectional propagation. Since $D_{PMJ}$ is not a true diffusion coefficient it may account at least partly for the need to scale by $K_1$ and $K_2$.

### Robin Conditions and Stimulus Currents in the Realistic Case

The code for the Robin conditions and the stimulus currents for the realistic Purkinje tree and myocardial meshes is as in the code above. What has been left out till now is how to find the PMJ vertices, labeled `stim_nodes`. For the test unit cube there was only one PMJ, and it was unimportant where it was placed. In the realistic tree used there are 157 ends that should be connected to the myocardium. In the Purkinje network there is one end node per PMJ, which is also the facet through which the flux flows between the two systems. In the myocardium there are several nodes per PMJ, and several exterior facets in some cases.

As mentioned in Section 5.5.1 the file `call_tree_msh.py` stores the indices of the end nodes in the Purkinje tree before refinement of the tree. The original Purkinje nodes are also nodes in the myocardial mesh. Since the myocardial mesh is not changed, the original indices of the end nodes are valid for the myocardial mesh. From this the corresponding vertex objects

can be found. Having the vertex object at a myocardial PMJ, all facets having this node as one of its own can easily be found by calling `facets(vertex object)`, and the same way with elements. Given a facet object it can be assessed whether it is an external facet or not by calling `facet.exterior()`.

The process of finding one facet per PMJ in the myocardium was the first step towards finding a bigger area. However, finding these facets proved to be a problem since the PMJs in some areas have such high density with respect to the mesh resolution that the facets overlapped. Overlapping areas means that one node would receive a stimulus from two or more Purkinje ends, which easily leads to divergence of the cell model solver (goss), and also is not physiologically representative. It was therefore decided that each myocardial end node should only belong to one PMJ area. Because of this only 109 of the 157 PMJs could be used. It would probably not be too hard to adjust the code so that some more facets could be found, but 109 facets was assumed to be enough on such a coarse grid. The Purkinje branches of the 48 excluded PMJs were not removed from the tree but simply not coupled to the myocardium neither through Robin conditions nor stimulus currents. In this way, the non-connected end branches are still visible in the plots, but as long as one is aware that there are 48 ends where no interaction should happen between the meshes, this should not be a problem.

With the two-way coupling for the Purkinje network and the myocardium implemented, anterograde and retrograde propagation was possible for the realistic meshes. Some parameter sets experienced divergence problems, but the final values from the test case, $K_1 = 3, K_2 = 2.945$ and $D_{PMJ} = 0.06$ cm$^3$/ms, with $D_P = 0.106$ cm$^2$/ms and $D_M = 0.013$ cm$^2$/ms worked well, giving quite realistic activation times for the Purkinje network and the myocardium.

# Chapter 6

# Results

During the implementation process a number of test cases were run to validate the implementation. The first section describes various test cases for the Purkinje network, while the second focuses on the coupling of the Purkinje network and the myocardial tissue. The last section shows some results from a simple implementation of left bundle branch block. The variable $x$ [cm] is the spatial coordinate in 1D.

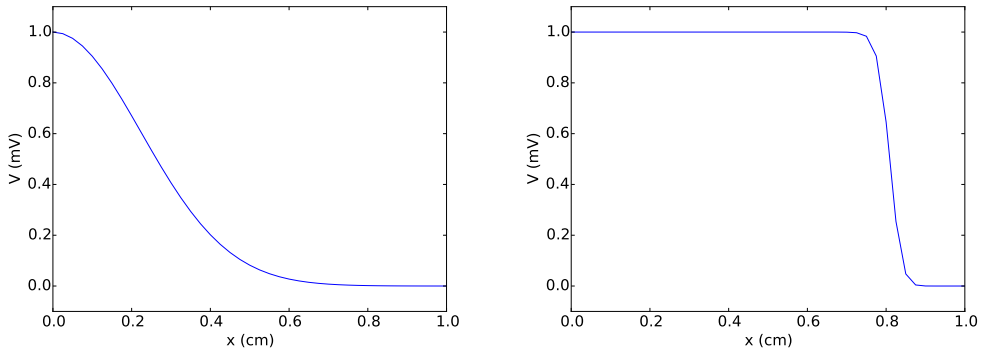
## 6.1 Tissue Model

### 6.1.1 First Test Case: The Bistable Equation

In section 3.1.1 it was said that a cubic polynomial is the simplest cell model that can reproduce the main behaviour of the depolarization phase of an action potential. When the cubic polynomial described is combined with the monodomain equation it is called the bistable equation.

This cell model consisting of a single ODE was implemented directly and stepped forward with the midpoint method. For the cubic polynomial it was stated that if the membrane potential was given a stimulus that makes it cross its threshold value $V_{th}$, it rises to $V_{peak}$, representing the depolarization phase, while if given only a small stimulus, the membrane potential will resturn to resting state $V_{rest}$. In space depolarization can spread with diffusion, resulting in a traveling wave front.

The bistable equation was tested with different values of $A$ and $V_{th}$ and different initial conditions on a unit interval domain. Some results for different values of $V_{th}$ are shown in Figure 6.1. The value 0.01 cm$^2$/ms for the diffusion coefficient $D$ gave wave fronts. If the diffusivity is too large, the wave will die out. With initial condition $V = e^{-10x^2}$ mV, see Figure 6.1a, the




**(a)** Initial condition

**(b)** $V_{th} = 0.1$, t=0.7

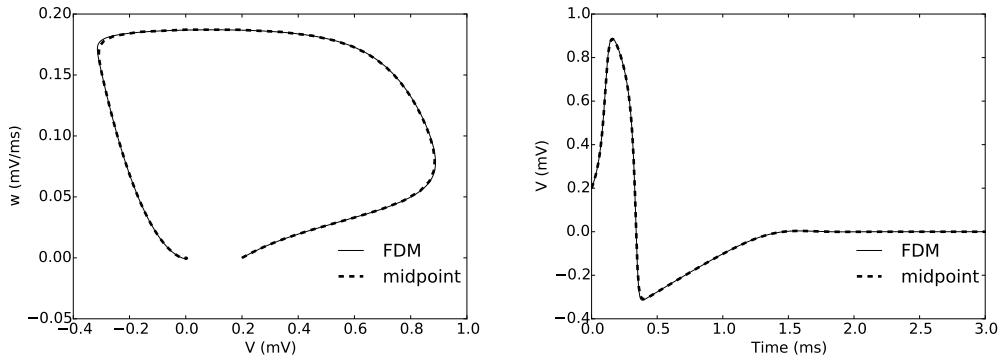
**Figure 6.1:** Membrane potential $V$ in the bistable equation on a unit interval at a chosen time (ms). The axes are not physiologically valid, as they both range from 0 to 1 mV in this test case. $A = 10$ mV$^{-1}$ ms$^{-1/2}$, $D = 0.01$ cm$^2$/ms, the unit interval consists of 40 elements and $\Delta t = 0.01$ ms. The initial condition in (a) is $V = e^{-10x^2}$ mV, (b) shows a traveling front.

initial values vary between 0 and 1 mV. The areas that have values above the threshold value will rise, while those with lower values will fall. For a low threshold value much of the area will be above the threshold, the values in this area will increase, and a wave front starts traveling towards the right. This can be seen in Figure 6.1b for $V_{th} = 0.1$ mV.

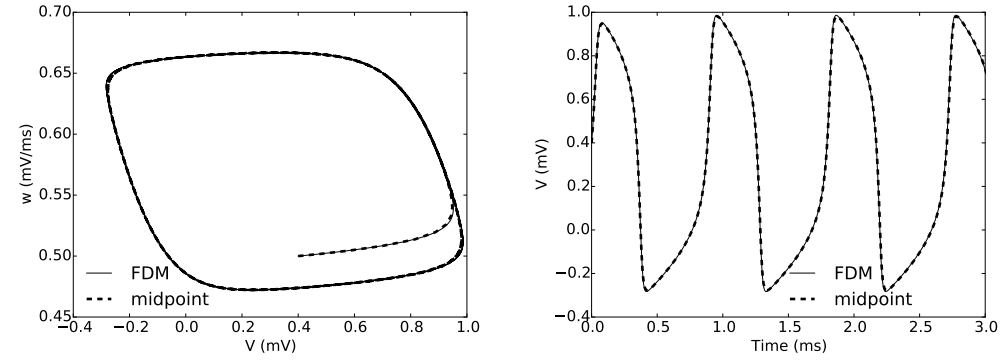
### 6.1.2 Validating the Midpoint Method: FHN Cell Model

To validate the implementation of the midpoint method, the monodomain equation was tested with no spatial diffusion ($D = 0$) and constant initial condition. Then all points should have the same value at all times, so a time series in an arbitrary point should equal the time series given by the non-spatial equations in the cell model. The FHN equations were used as cell model. When $D = 0$, the linear PDE reduces to $V^{n+1} = V^n$, so the only change comes from the system of ODEs, which are here stepped forward using the midpoint method. The cell model equations in the non-spatial model were approximated by a finite difference scheme.

With the parameter values used by Keener and Sneyd [14], the resulting plots can be seen in Figure 6.2. The two schemes are almost identical, and closely resemble the relevant plots in Keener and Sneyd [14, Figs. 5.15 and 5.16]. Because of this it is reasonable to assume that the finite difference scheme and the midpoint method are both correctly implemented.




**(a)** $I_{app} = 0$ **(b)** $I_{app} = 0.$




**(c)** $I_{app} = 0.5.$ **(d)** $I_{app} = 0.5.$

**Figure 6.2:** Phase portrait for the FHN equations (3.8) - (3.10) for the finite difference method (FDM) given by equations (4.12) - (4.13) and an arbitrary point in a finite element method (FEM) for the monodomain equation with midpoint method, with FHN cell model, no spatial diffusion ($D = 0$ cm$^2$/ms) and constant initial condition. $A = 1$ mV$^{-1}$ ms$^{-1/2}$, $\alpha = 0.1$ mV, $\gamma = 0.5$ and $\epsilon = 0.01$. In (a) and (b) $I_{app} = 0$ $\mu$A/$\mu$F and initial conditions are $V = 0.2$ mV, $w = 0.0$ $\mu$A/$\mu$F. In (c) and (d) $I_{app} = 0.5$ $\mu$A/$\mu$F and initial conditions are $V = 0.4$ mV, $w = 0.5$ $\mu$A/$\mu$F. $\Delta t = 0.0001$ ms for FDM. $\Delta t = 0.01$ ms for FEM, and the spatial area is a unit interval with 100 elements.

### 6.1.3 Validating the Goss-Solver: FHN Cell Model

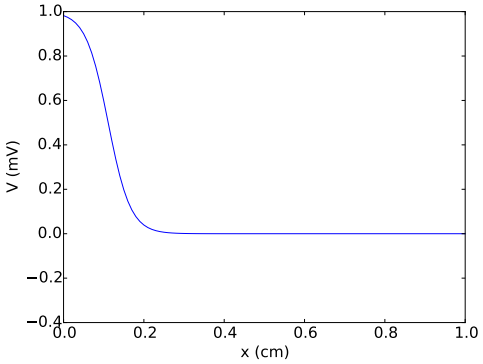
To validate the implementation of the goss-solver, results using the goss-solver and the midpoint method were compared. The cell model used was a slightly adjusted FHN model.

When the midpoint method was used, the ODEs were explicitly implemented. To use a goss-solver, the ODEs had to be declared in gotran format, and a solver had to be defined. There is one cell model for the whole domain. The gotran file looks as follows:

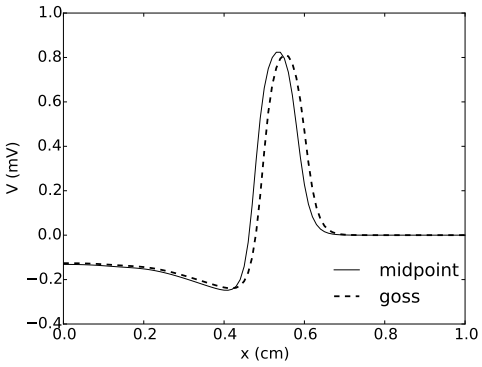
```
# Cell model file for the modified FitzHugh-Nagumo model

# States with default initial conditions
states(v=0.5,
       w=0.5)

# Parameters
eps = 0.01; I_stim = 0.5; alpha = 0.1; gamma = 0.5

# Time derivatives
dv_dt = (-v*(v - alpha)*(v - 1) - w + I_stim)/eps
dw_dt = 0.8*(v - gamma*w)
```

Different values were tested for the spatial diffusion and initial conditions. The initial condition $V = 0.5\,(1 - \tanh(18x - 2))$ mV, which can be seen in Figure 6.3a, is a traveling wave profile on the same form as in Keener and Sneyd [14, eq.(6.19)], but adjusted with respect to placement, traveling direction and steepness of the wave front.

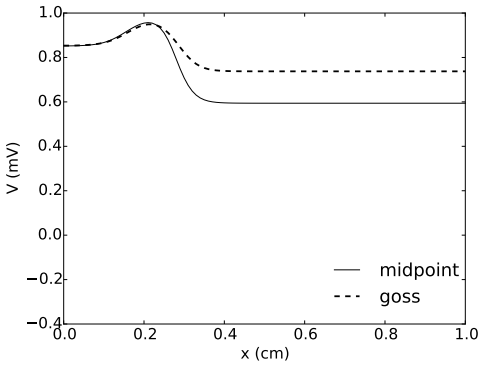
With initial condition $w = 0$ $\mu$A/$\mu$F, $I_{app} = 0$ $\mu$A/$\mu$F, and the right hand side of equation (3.9) for $w$ multiplied by 0.8 to lessen the dampening effect of $w$, the result is a traveling pulse, see Figure 6.3b. If equation (3.9) is not adjusted, the traveling pulse will die out. For $I_{app} \neq 0$, a more complicated picture arises, since $I_{app}$ is applied to all points in the domain. The result, which can be seen in Figure 6.3c for $I_{app} = 0.5$ $\mu$A/$\mu$F and initial condition $w = 0.5$ $\mu$A/$\mu$F, is some kind of waves. The initial wave front is traveling towards the right, while at the same time, the whole solution grows because of the stimulation from $I_{app}$. The results from the midpoint method and the goss-solver are similar, but the goss-solver seems to make the wave move a bit faster.




**(a)** Initial condition of $V$.




**(b)** $I_{app} = 0$, t = 1.0.



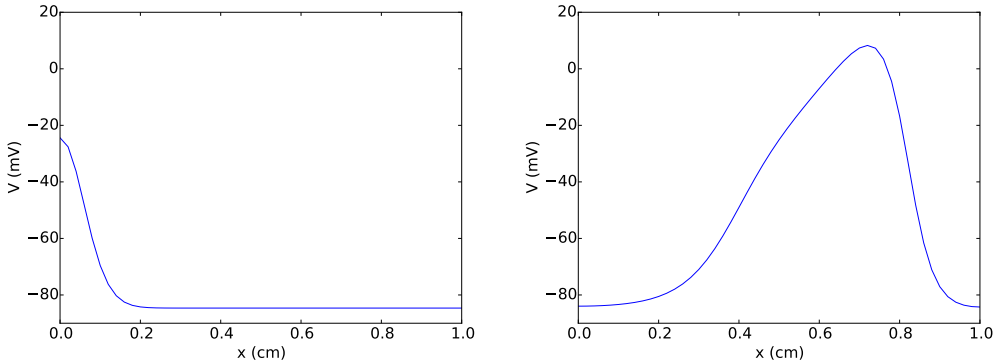

**(c)** $I_{app} = 0.5$, t = 0.2.

**Figure 6.3:** Membrane potential in space for the monodomain equation with midpoint method and goss-solver, with slightly adjusted FHN cell model. Equation (3.9) is multiplied by 0.8. The spatial area is a unit interval with 100 elements. The spatial diffusion coefficient $D = 0.01$ cm$^2$/ms, $\Delta t = 0.01$ ms, $A = 1$ mV$^{-1}$ ms$^{-1/2}$, $\alpha = 0.1$ mV, $\gamma = 0.5$ and $\epsilon = 0.01$. (a) shows the initial condition of $V$: $\frac{1}{2}(1 - \tanh(18x - 2))$ mV, (b) shows a traveling pulse resulting from $I_{app} = 0$ $\mu$A/$\mu$F and initial condition $w = 0.0$ $\mu$A/$\mu$F. In (c) $I_{app} = 0.5$ $\mu$A/$\mu$F and initial condition for $w$ is 0.5 $\mu$A/$\mu$F.

### 6.1.4 Testing the Modified FHN Cell Model

Here the FHN cell model modified by Rogers and McCulloch, as referred by Sundnes et al. [24, p.35] is studied. In goss there is already a gotran file for this [12]. Only a part of the domain will be stimulated. The cell model that is not going to be stimulated is created by just removing the stimulus term from the model with stimulus. The goss-solver is created in the way described in Section 5.3.2.

In Figure 6.4 two images are shown at different times of the growing and then traveling pulse for diffusion factor $D = 0.0001$ cm$^2$/ms. If the diffusion is too low, the wave don't move, and if it is too high, the wave dies out. To find a diffusion factor that gives a traveling wave, different values were tested, and the resulting animations studied.




**(a)** t = 30 **(b)** t = 400

**Figure 6.4:** Membrane potential in space for the monodomain equation with goss-solver, with modified FHN cell model at two chosen points in time. The spatial area is a unit interval with 50 elements. The area to the left, where x $<$ 0.05 cm receives a stimulus, while the rest of the area does not, resulting in a wave traveling from left to right. The spatial diffusion coefficient $D = 0.0001$ cm$^2$/ms, $\Delta t = 1.0$ ms. Initial conditions are $V = -84.058830$ mV, $w = 0.0$ $\mu$A/$\mu$F.

The time series in $x = 0.9$ cm is shown in Figure 6.5. The point is arbitrary, the only thing that should vary for different points is the time. This plot is of similar shape and time scale as Figure 2.3 in Sundnes et al. [24], which is for a single cell, indicating that the model is correctly implemented, but the shapes are not identical. The upstroke velocity in Figure 6.5 is lower than that in Figure 2.3 Sundnes et al. [24], probably as a result of different stimuli and because of diffusion.

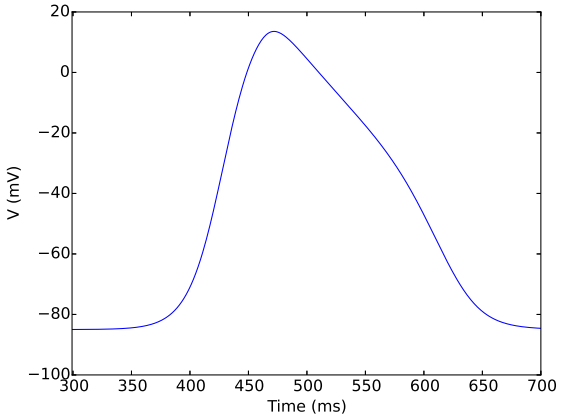



**Figure 6.5:** Membrane potential in time in the spatial point $x = 0.9$ cm for the case described in Figure 6.4.

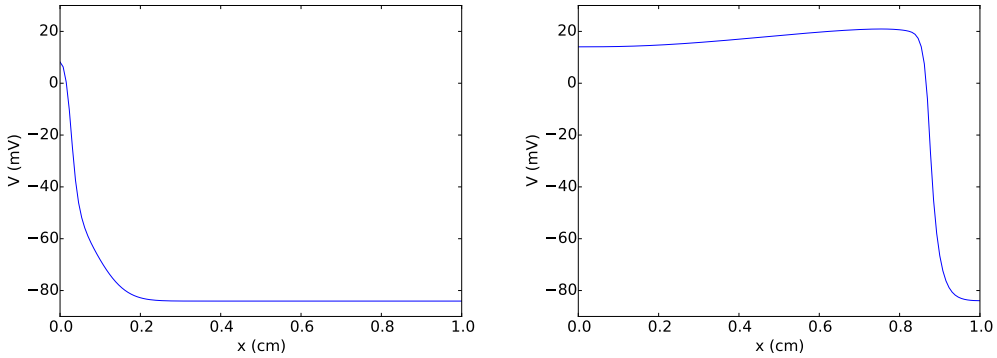
### 6.1.5 Testing the PRd Cell Model

#### Unit Interval

The PRd model was run with a stimulus on part of the domain. Different diffusion coefficients were tested to find some values that give a traveling wave. In Figure 6.6 there are two images at different times of the growing and then traveling wave, in the same way as was done for the modified FHN model in Figure 6.4. This wave returns to its resting state much slower.

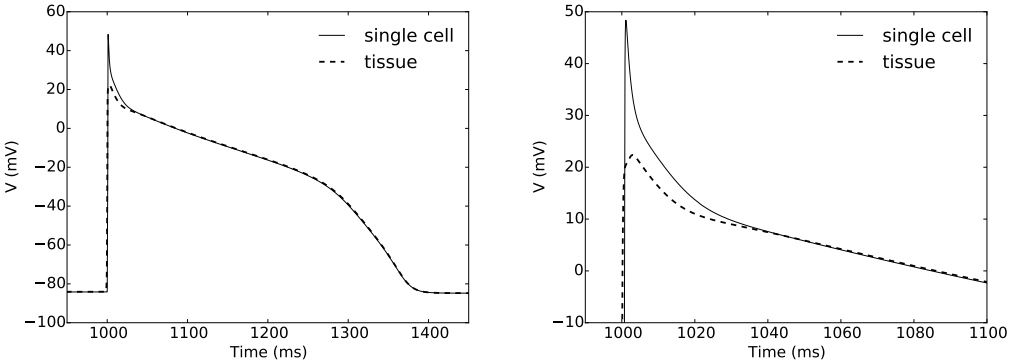
The time series in a point is shown in Figure 6.7, together with the time series of the original PRd model in C code. One can see that the shape of the two action potentials are the same, but the original model has a higher peak. A possible reason for this could be that the C code has a dynamic time step, while the monodomain solver does not. A dynamic time step will decrease its step length when changes are rapid, as they are in the depolarization phase, giving a more correct wave profile. The monodomain solver might not capture all the rapid changes.

However, Decker et al. [15] also get lower action potential peak for tissue than for single cells in their model, and this has also been observed in several experimental studies referred by Decker et al. This is due to the electrotonic load during propagation. The results in Figure 6.7 are therefore as should be expected, in accordance with observations and other studies.




**(a)** t = 2.2 **(b)** t = 16.0

**Figure 6.6:** Membrane potential in space for the monodomain equation with goss-solver, with PRd cell model at two chosen points in time. The spatial area is a unit interval with 130 elements. The area to the left, where x $<$ 0.05 cm receives a stimulus, while the rest of the area does not, resulting in a wave traveling from left to right. The spatial diffusion coefficient $D = 0.001$ cm$^2$/ms, $\Delta t = 0.1$ ms. Initial condition is $V = -84.058830$ mV.




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

**Figure 6.7:** Membrane potential in time for the PRd model on a single cell (C code) and on a unit interval tissue (gotran code). The graph for the tissue is in the spatial point $x = 0.9$ cm for the case described in Figure 6.6. The tissue graph is given an offset to make the two graphs overlap as much as possible, as the starting point is unimportant. (b) is a detail from (a).

### Small Trees

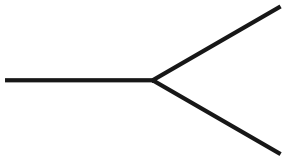
Solving the monodomain equation with the PRd cell model on the mesh in Figure 6.8a, where each branch has length 1 cm, gives the rest of the plots in Figure 6.8. The plotting used is in 1D, which is obviously not fit for this purpose, but it was only used as a first test to see if the results were reasonable.

The mesh consists of three branches of equal length and 120 elements. The 40 first elements make up the first branch, the next 40 the upper right branch and the last 40 the bottom right branch. The angle between the right branches is 60 degrees.

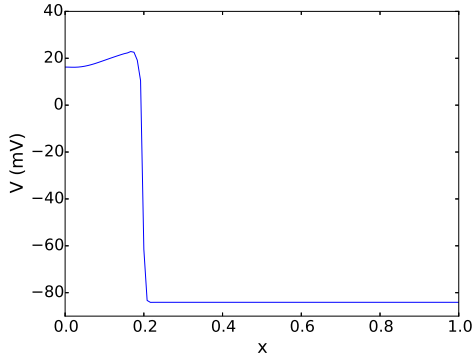
The wave starts out normally, but when it has moved 1/3 of the length of the mesh, a new wave pops up at 2/3 of the length. This corresponds to the wave reaching the branching point, splitting into two waves, one in each of the two right branches. The wave in the upper branch looks like a continuation of the original wave, while the lower branch make up the last 1/3 of the mesh, which is where wave number two pops up.

Next another branching point was added, as can be seen in Figure 6.9a. Here the elements of the first single branch, the upper branch, and the first half of the lower branch are as in the previous case. The lower branch is then split in two, where the upper part has elements 101 to 110, and the lowest part has element numbers 111 to 120. From this the wave is expected to start off as before: first there is one wave, when this wave reaches 1/3 of the length of the mesh it should continue together with a new wave popping up at 2/3 of the length of the mesh. This last wave is the wave in the lower branch, so when this has gone 1/2 of its stretch it splits in two, the one in the upper branch looking as a continuation, the one in the lower branch popping up at the last 1/4 of the last 1/3 of the mesh. This is exactly what happens, see Figure 6.9.

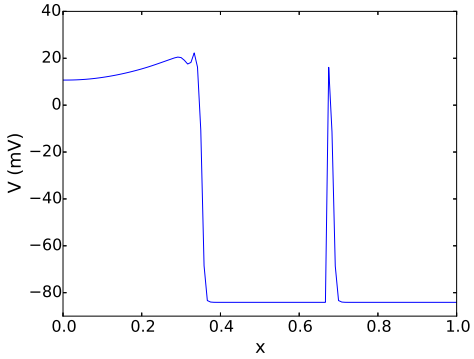
In the top and bottom rows of Figure 6.10 the signal propagation through another small test tree is shown in a more appropriate way. The tree consists of the first five cables in the Purkinje network.



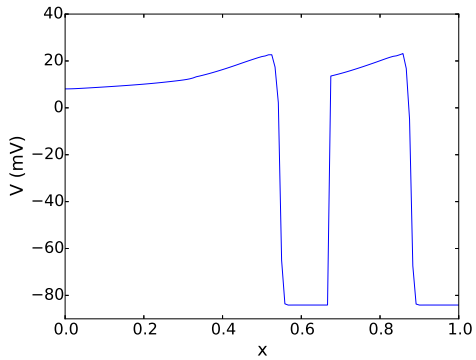
**(a)** Test tree



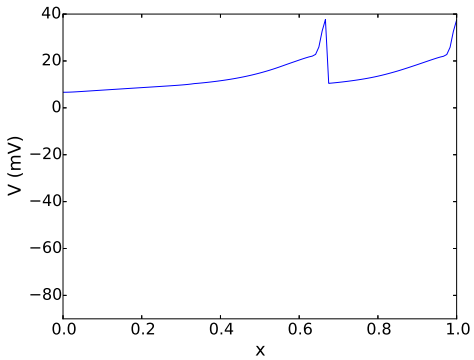

**(b)** t = 13.9




**(c)** t = 25.2

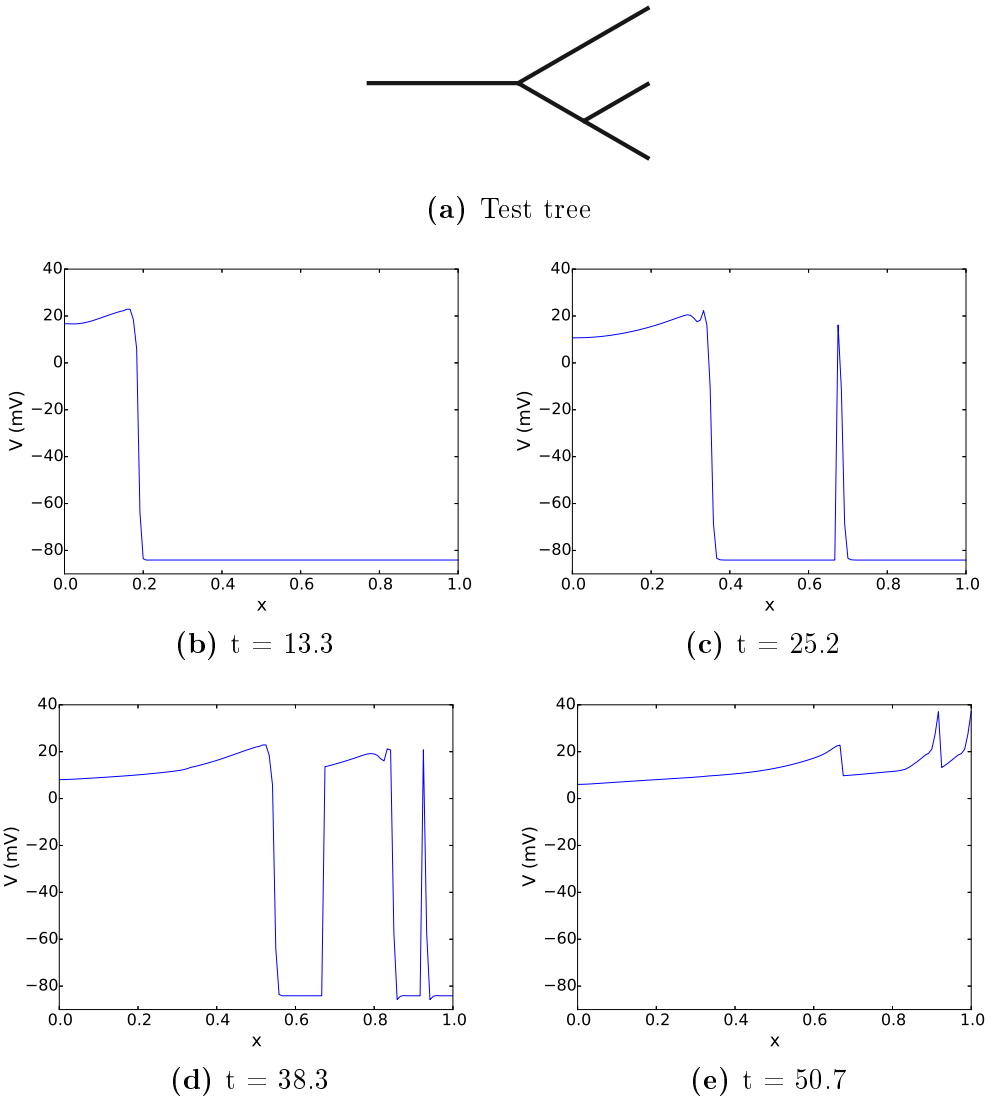



**(d)** t = 38.3




**(e)** t = 47.1

**Figure 6.8:** Membrane potential in space for the PRd cell model at four chosen points in time (ms). The mesh is the one in Figure 6.8a, where each branch has length 1 cm. The mapping of the branches to the unit interval is described in the text. The area to the left in the mesh, where $x < 0.05$ cm receives a stimulus, while the rest of the area does not. Since the mesh is mapped to 1D, the spatial variable $x$ in the plots is scaled, so it is not measured in cm. $D = 0.0005$ cm$^2$/ms, $\Delta t = 0.1$ ms. Initial condition is $V = -84.058830$ mV.




**(a)** Test tree

**(b)** t = 13.3

**(c)** t = 25.2

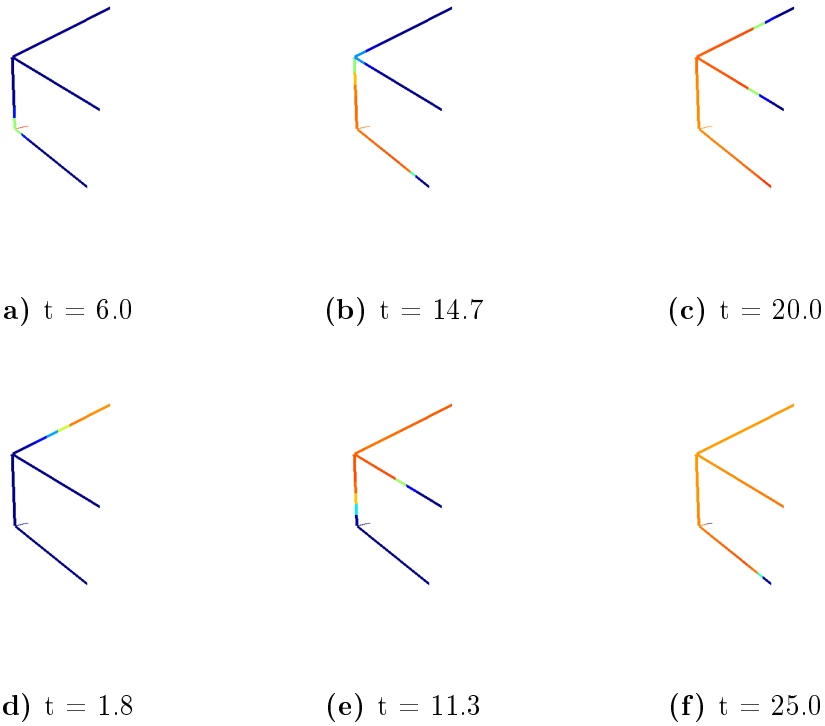
**(d)** t = 38.3

**(e)** t = 50.7

**Figure 6.9:** Membrane potential in space for the monodomain equation with goss-solver, with PRd cell model at four chosen points in time (ms). The mesh is the one in Figure 6.9a, where the leftmost branch has length 1 cm and the others have lengths as seen relative to this. The mapping of the branches to the unit interval is described in the text. The area to the left, where $x < 0.05$ cm receives a stimulus, while the rest of the area does not. Since the mesh is mapped to 1D, the spatial variable $x$ in the plots is scaled, so it is not measured in cm. $D = 0.0005$ cm$^2$/ms, $\Delta t = 0.1$ ms. Initial condition is $V = -84.058830$ mV.

### 6.1.6 Signal Propagation in the Opposite Direction

Here the mesh consisting of the first five cables in the Purkinje network, as shown in Figure 5.2a, was used. The top row in Figure 6.10 shows the propagation through the tree when the first branch was stimulated. The bottom row shows the propagation when a different area ($y > 2.6$ cm) was stimulated. A different stimulus area was chosen to check whether a signal could travel both ways at junctions, which it did, giving both anterograde and retrograde propagation.



**(a)** t = 6.0 **(b)** t = 14.7 **(c)** t = 20.0

**(d)** t = 1.8 **(e)** t = 11.3 **(f)** t = 25.0

**Figure 6.10:** Membrane potential in space for the monodomain equation with goss-solver, with PRd cell model at chosen points in time (ms). The mesh is the one in Figure 5.2a. $D = 0.0008$ cm$^2$/ms, $\Delta t = 0.1$ ms. Initial condition is $V = -84.058830$ mV. The colorscale ranges from -85 mV (dark blue) to 30 mV (red). The top row shows the signal starting in the first branch. The first cable receives a stimulus, while the rest of the area does not. The bottom row shows the signal starting in an other branch than the first, giving retrograde propagation. The area where $y > 2.6$ cm receives a stimulus, while the rest of the area does not.

### 6.1.7 Realistic Tree

With the signal propagating correctly through a test tree, both in the anterograde and retrograde directions, the next step was to take the realistic Purkinje tree to the test. To get propagation, different diffusion coefficients were tested as before until some suitable values were found. For the FHN model this testing stopped as soon as a diffusivity that gave propagation was found. Plots of the propagation is shown at four time steps in Figure 6.11.

For the PRd model it was desirable to achieve realistic activation times for the Purkinje tree. In Chapter 2 it was stated that it took approximately 25 ms from the stimulation of the bundle of His until the whole Purkinje tree was activated. The fastest activation time achieved in this thesis was 27.8 ms, which is within reasonable range from the desired value. This was achieved for a tissue diffusivity of 0.106 $\text{cm}^2/\text{ms}$. Four time steps are shown in Figure 6.12.

What would have been more interesting to know than the diffusivity is the conductivity, so it could be compared to other cases. The diffusion coefficient is defined as $D = \sigma/(\chi C_m)$, where $\sigma$ is the conductivity. The PRd cell model defines $C_m = 1\ \mu\text{F}/\text{cm}^2$. With a value for $\chi$, the conductivity could be found. In Sundnes et al. [24, Table 2.1], $\chi = 2000\ \text{cm}^{-1}$. This might not hold here, but it could still be interesting to see what conductivity this would yield:

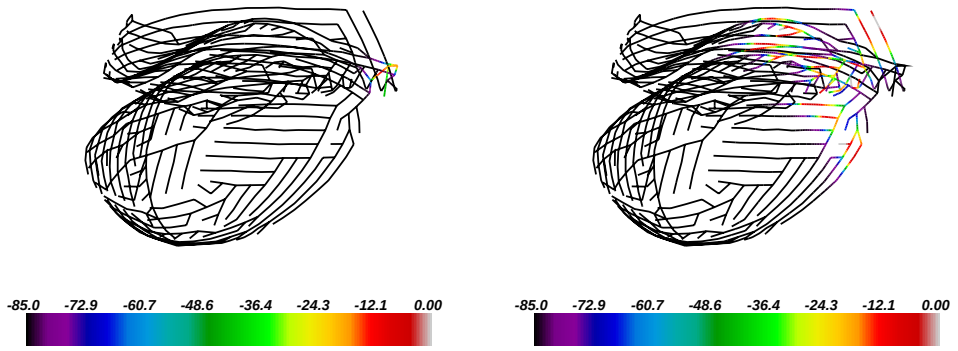
$D = 0.106\ \text{cm}^2/\text{ms} = \sigma\ \text{cm}/2000\ \text{cm}^2/\mu\text{F}$.

$\Rightarrow \sigma = 2000 \cdot 0.106\ \mu\text{F}/(\text{cm ms}) = 212\ \text{mS/cm}$.

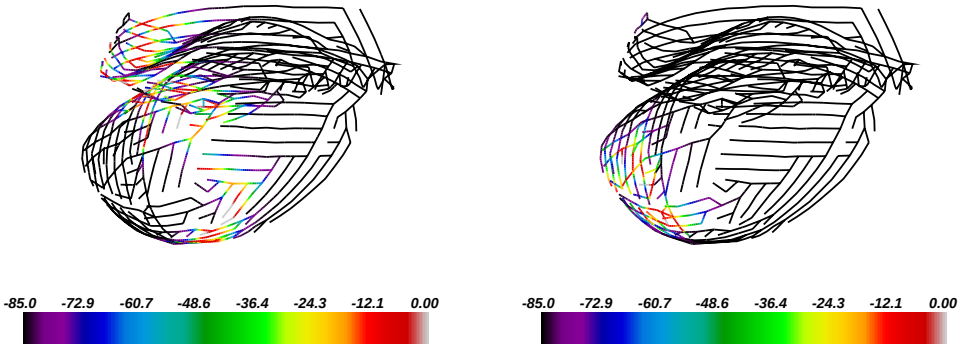
Deo et al. [6] operate with an intracellular conductivity in the Purkinje tree of 15 mS/cm, so 212 mS/cm is very high, but realistic conductivity values cannot be expected unless a very fine grid is used.

Comparing the plots for the FHN and the PRd cell models, at first sight it might seem like the FHN action potential is very short compared to the PRd model. Returning to Figures 3.2b and 3.3 one can see that the modified FHN cell model has an action potential duration of about 250 ms, while the PRd model has an action potential duration of about 400 ms. This is a significant difference, but not enough to explain the differences in the plots of propagation through the tree. The explanation comes from looking at the time scale. The FHN wave front moves very slowly through the tree compared to the PRd wave.

Figure 6.12e shows the action potentials of four points in the Purkinje tree. The points are shown in Figure 6.12f. The first point is in the beginning of the tree. This node gets a stimulus current, so its upstroke looks a bit different from the others. The next point is somewhere in the middle of the tree, and is a "normal" action potential. The last two action potentials are from end nodes in the tree. They both have higher upstroke amplitude than

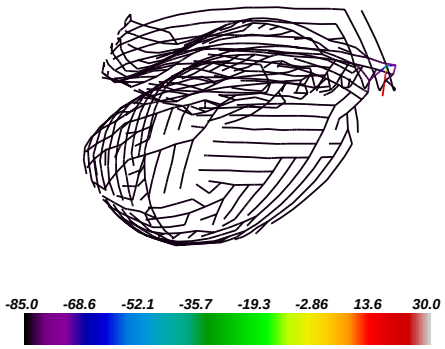



**(a)** t = 150 **(b)** t = 990

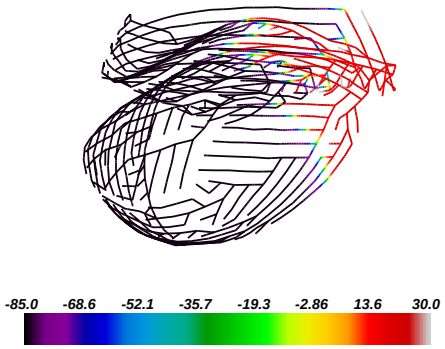



**(c)** t = 2010 **(d)** t = 2640

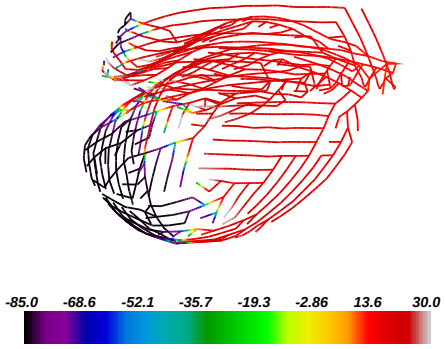
**Figure 6.11:** Membrane potential in the Purkinje network for the monodomain equation with goss-solver, with FHN cell model at four chosen points in time (ms). The first branch and a small area close to it receives a stimulus while the rest of the area does not. `ds` = 0.05 cm, $D = 0.001$ cm$^2$/ms, $\Delta t = 30$ ms. Initial condition is $V = -84.058830$ mV.



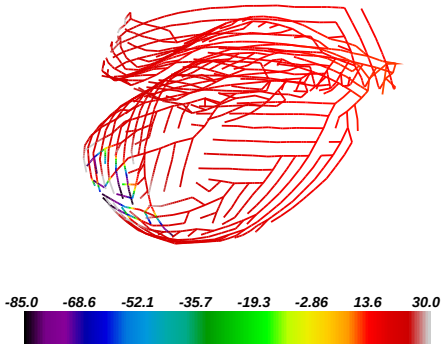

**(a)** t = 4.5



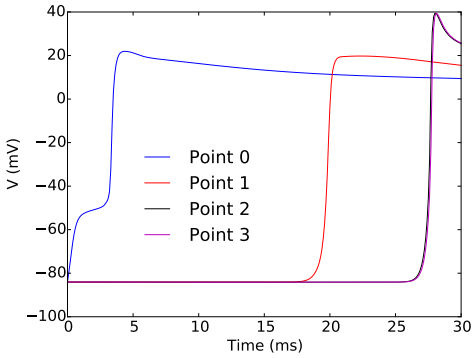

**(b)** t = 13.6



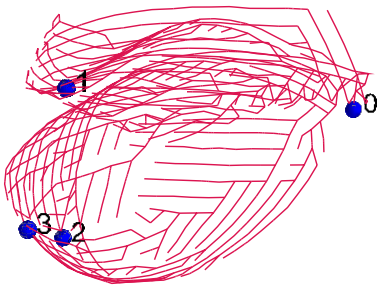

**(c)** t = 22.1




**(d)** t = 26.1




**(e)** PRd




**(f)** Timeseries points

**Figure 6.12:** Membrane potential in the Purkinje network for the monodomain equation with goss-solver, with PRd cell model at four chosen points in time (ms) (a)-(d), and action potentials in four points in space (e)-(f). The first branch and a small area close to it receives a stimulus while the rest of the area does not. `ds` = 0.05 cm, $D = 0.106$ cm$^2$/ms, $\Delta t = 0.1$ ms. Initial condition is $V = -84.058830$ mV.

the two first points. This is due to the homogenuous Neumann conditions on all ends in the Purkinje network. They make the ends in the tree electrically insulated, so the signal arriving has nowhere to "escape" fast enough to keep the amplitude to its normal height. This effect should disappear when the Purkinje tree is properly coupled to the myocardium because of electrotonic loading from the myocardial mass.

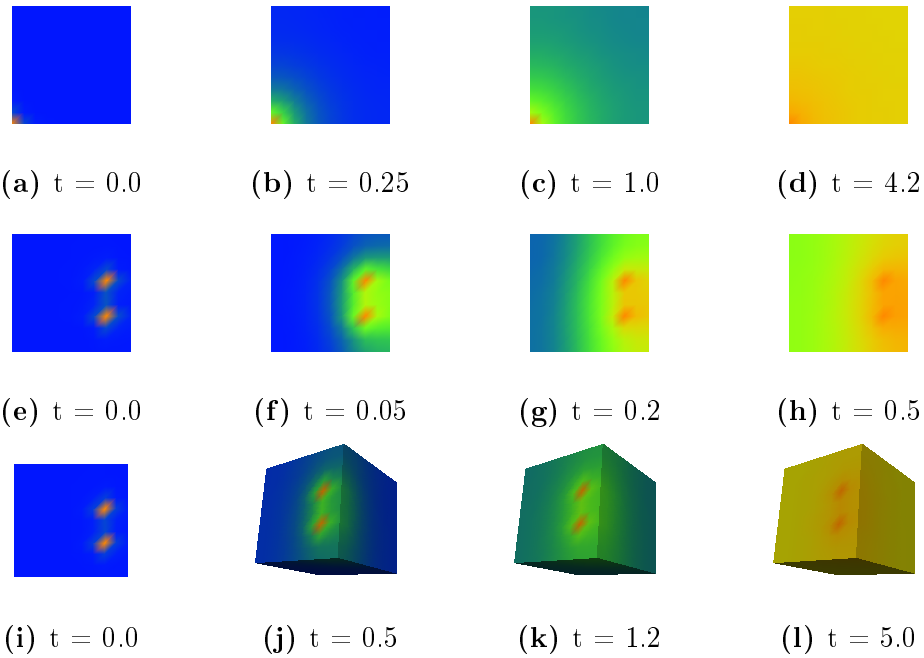
## 6.2 Coupling the Purkinje Network and the Myocardial Tissue

### 6.2.1 One-Way Coupling

The first test case was a unit square connected to the end of a Purkinje tree in a single point in the lower left corner of the square. The initial condition is $V = -84.058830$ mV on the whole area, while the stimulus is simply kept at 0 mV. It is expected that a wave will spread from the corner that is connected to the Purkinje tree to the rest of the area, falling to rest when the whole area has transmembrane potential equal to zero. This is exactly what happens, as can be seen in the top row of Figure 6.13, where the time increases from left to right.

Next two ends of the Purkinje network were connected to a unit square, see the middle row in Figure 6.13, and and a unit cube, see Figure 6.13, bottom row. This worked well, giving wave propagation as expected, so the same coupling method was tested on the realistic Purkinje network and myocardial tissue meshes. In the test case the myocardial diffusivity $D_M$ was set to 1.0 cm$^2$/ms. This proved too high for the realistic mesh. The diffusivity $D_M = 0.01$ cm$^2$/ms gave activation of the myocardial tissue, the membrane potential in space is shown for both the Purkinje tree and the myocardium at three time steps in Figure 6.14.

The total activation time of the ventricles was about 84 ms, which is a bit high, but this will be better adapted for the model that can conduct the signal both ways at the PMJs.

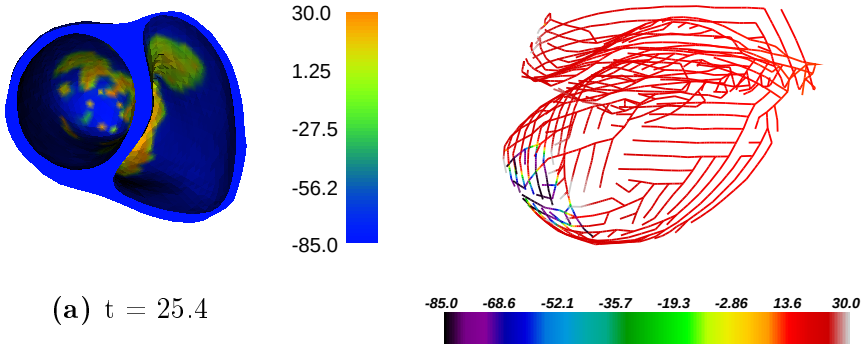


**(a)** t = 0.0 **(b)** t = 0.25 **(c)** t = 1.0 **(d)** t = 4.2

**(e)** t = 0.0 **(f)** t = 0.05 **(g)** t = 0.2 **(h)** t = 0.5

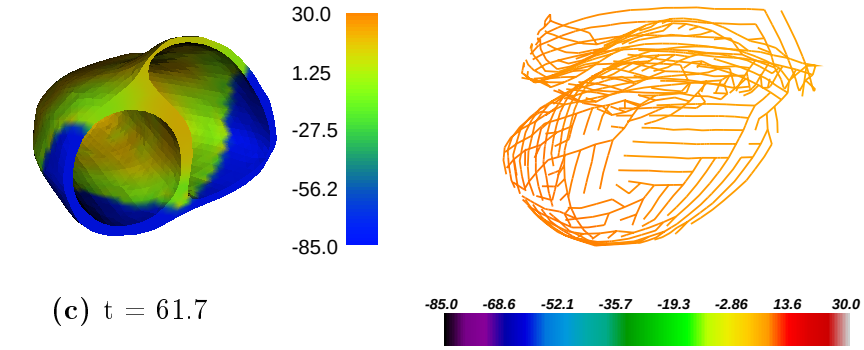
**(i)** t = 0.0 **(j)** t = 0.5 **(k)** t = 1.2 **(l)** t = 5.0

**Figure 6.13:** Membrane potential in space for the diffusion equation. All times are in ms. $D_M = 1.0$ cm$^2$/ms, $\Delta t = 0.01$ ms. Initial condition is $V = -84.058830$ mV. A pointwise Dirichlet boundary condition of $V = 0.0$ mV is enforced in specified points. The colorscale ranges from $-85$ mV (dark blue) to 0 mV (orange). Top row: the domain is the unit square with $10 \times 10$ elements, Dirichlet condition is enforced in point $(0.0, 0.0)$, i.e. the bottom left corner. Middle row: the domain is the unit square with $10 \times 10$ elements, Dirichlet conditions are enforced in points $(0.8, 0.3)$ and $(0.8, 0.6)$. Bottom row: the domain is the unit cube with $10 \times 10 \times 10$ elements, Dirichlet conditions are enforced in points $(0.8, 0.3, 1.0)$ and $(0.8, 0.6, 1.0)$.



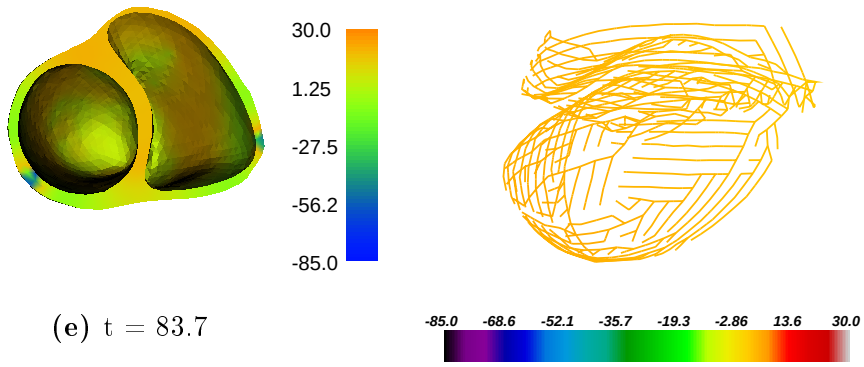

(a) t = 25.4

(b) t = 25.4




(c) t = 61.7

(d) t = 61.7




(e) t = 83.7

(f) t = 83.7

**Figure 6.14:** Membrane potential in the Purkinje network (right) and the myocardium (left). The Purkinje network uses the PRd cell model with $D_P = 0.106$ cm$^2$/ms, while the myocardium uses the Decker et al. cell model with $D_M = 0.01$ cm$^2$/ms. The first branch and a small area close to it in the Purkinje network receives a stimulus while the rest of the area does not. The end values in the Purkinje network are used as Dirichlet boundary conditions for the myocardium. Initial condition is $V = -84.058830$ mV for both models and `ds` $= 0.05$ cm, $\Delta t = 0.1$ ms. All times are in ms. The colorscales have units mV.

### 6.2.2 Two-Way Coupling

#### Robin Boundary Conditions and Stimulus Currents

In the test case for Robin conditions on the Purkinje tree and stimulus currents for the myocardium, a unit cube consisting of $10 \times 10 \times 10$ elements represented the myocardium and the first four branches in the realistic Purkinje tree, with element lengths of about 500 $\mu$m, represented the Purkinje tree. These tissues had one PMJ between them.

Many parameter sets were tested. The overall tendency seemed to be that without both scaling up the stimulus current by a factor $K_2$ and increasing the number of myocytes to receive a stimulus from the original three myocyte nodes on a facet, anterograde propagation was impossible. With the scaling factor $K_2$ present, it seemed impossible to get retrograde activation without introducing a scaling factor $K_1$ for the Robin condition. In addition $D_{PMJ}$ must make sure that the Purkinje tree is not drained too fast so the signal dies out in the tree, and the myocardium not filled too slowly, so the signal diffuses out in the myocardial tissue before activation can happen.

To study the effect of the number of myocyte nodes to receive a stimulus, a stimulus was started in the myocardial tissue, varying the strength of the stimulus and the number of nodes. It was found that if the number of nodes was very small the activation died out locally, not managing to activate the rest of the tissue, no matter how strong the stimulus was. For a stimulus area a bit larger, the activation could spread to the rest of the tissue given a strong enough stimulus. If the stimulus area consisted of quite many nodes it would activate the whole tissue at much lower stimuli strengths. It seemed that with more nodes receiving a stimulus, the nodes at the edge of the stimulus area act as a buffer between the nodes in the middle of the stimulus area and the nodes outside the stimulus area. In this way, the nodes in the middle of the area have time to build their action potential higher than they would have been able to if their immediate neighbouring nodes were draining them fully. When these middle nodes have reached a high enough potential, an activation wave starts to spread from them through the rest of the tissue.

Extending the original stimulus area of three myocyte nodes to the fourth node of the element with the same facet, all nodes in elements bordering to the first element were added. This process can be repeated as many times as necessary. In this way, each extension round would add another "layer" of cells to a sort of hemisphere with the original facet in the center. In the article by Deo et al. [7], the PMJ areas consist of a single node in the Purkinje tree and all the myocyte nodes within a radius of 500 $\mu$m from the Purkinje node. The average number of myocyte nodes coupled to a Purkinje

node is 56. Because the myocardial mesh used in this thesis is quite coarse, a volume much greater than 500 $\mu$m in radius must be stimulated to come close to 56 nodes.

Table 6.1 shows how a higher value of the scaling factor $K_2$ for the stimulus currents allows for a smaller number of stimulus nodes to still achieve propagation in both directions for the scaling factor for the Robin condition $K_1 = 3$. The "extension levels" result from the process described above of adding a layer of cells to the group of initial elements. Extension level 0 consists of all nodes in all cells where the original node is a node, then the level increases with one for each added cell layer.

| Extension level | Num nodes | Num cells | $K_2$ | Activation |
|---|---|---|---|---|
| 0 | 11 | 6 | 13 | − |
| | | | 14 | + |
| 1 | 42 | 96 | 2 | − |
| | | | 3 | + |
| 2 | 106 | 324 | 1 | − |
| | | | 2 | + |
| 3 | 215 | 768 | 1 | − |
| 4 | 375 | 1484 | 1 | + |

**Table 6.1:** Overview over how the extension level and $K_2$ work together. A "+" means that this parameter set gives both anterograde and retrograde propagation, while "−" means that propagation in at least one of the directions fails. $K_1 = 3$, $D_{PMJ} = 0.06$ cm$^3$/ms, $D_P = 0.106$ cm$^2$/ms and $D_M = 0.01$ cm$^2$/ms

Next, Table 6.2 shows how a higher value of $K_1$ allows for a smaller value of $K_2$ to achieve bidirectional propagation. In Table 6.1 it can be seen that the combination $K_1 = 3$, $K_2 = 2$ at extension level 1 did not give bidirectional propgation. If $K_1$ is increased to 7 one gets bidirectional propgation, as shown in Table 6.2. No value for $K_1$ could be found that gave propagation both ways for $K_2 = 1$ at extension level 1.

In the same way, Table 6.1 also shows that the combination $K_1 = 3$, $K_2 = 13$ at extension level 0 did not give bidirectional propgation. If $K_1$ is increased to 10 one gets bidirectional propgation, as shown in Table 6.2. For $K_1 = 10$, $K_2$ can be lowered to 9 and this will still give propagation both ways. For $K_2 = 8$, a value of 15 or higher for $K_1$ is needed to give bidirectional propgation.

However, this tendency of higher $K_1$ values allowing for smaller $K_2$ values

does not hold forever. Too high values for $K_1$ will drain the Purkinje tree too much too fast, and again, there is no activation of the myocardium.

| Extension level | $K_2$ | $K_1$ | Activation |
|---|---|---|---|
| 1 | 2 | 7 | + |
| | 2 | 6 | − |
| | 1 | x | x |
| 0 | 13 | 10 | + |
| | 13 | 9 | − |
| | 9 | 10 | + |
| | 8 | 10 | − |
| | 8 | 15 | + |
| | 8 | 14 | − |

**Table 6.2:** Overview over how $K_1$ and $K_2$ work together. A "+" means that this parameter set gives both anterograde and retrograde propagation, while "−" means that propagation in at least one of the directions fails. The "x" means that for $K_2 = 1$, no value for $K_1$ was found that gave propagation in any direction. $D_{PMJ} = 0.06$ cm$^3$/ms, $D_P = 0.106$ cm$^2$/ms and $D_M = 0.01$ cm$^2$/ms.

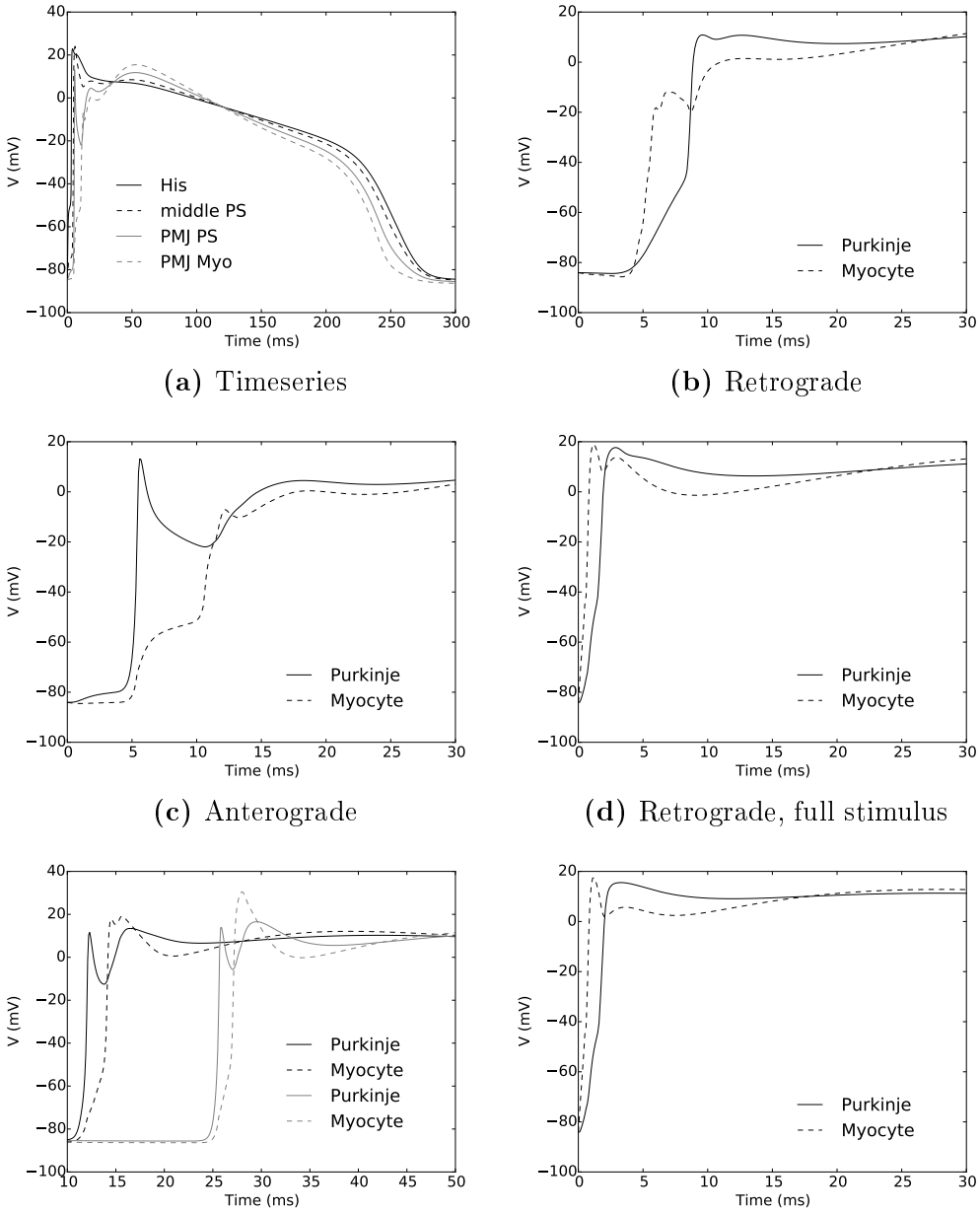
Now it might seem that the best choice of parameters is simply a relatively high value for both $K_1$ and $K_2$ (but not so high that the Purkinje tree becomes too drained) as this gives both anterograde and retrograde propagation for a small stimulus area in the myocardium. The problem with this, in addition to the fact that it would have been preferable not to need these factors at all, so high values are not desirable, is that high values of both $K_1$ and $K_2$ diminsh PMJ delay times.

Of the boundary conditions giving a two-way connection between the Purkinje network and the myocardial tissue, only Robin conditions for the Purkinje problem and stimulus currents for the myocardial problem gave any noticeable delay for anterograde and retrograde propagation. Time of activation for a node was taken to be the time of the steepest upstroke. Delay times at PMJs were defined as the difference between the activation time of the single Purkinje node and the average of the activation times for the coupled myocyte nodes. Lower values of each of the factors $K_1, K_2$ and $D_{PMJ}$ gave a higher PMJ delay, but too low values of any of them, and bidirectional propagation was lost. It was not easy to find a clear pattern for the influence of $K_1$ and $K_2$, but the best delay times came from giving them quite similar and low values.

The values found that gave the highest anterograde delay for the unit cube were $K_1 = 3, K_2 = 2.945$ and $D_{PMJ} = 0.06$ cm$^3$/ms when $D_P = 0.106$ cm$^2$/ms, $D_M = 0.01$ cm$^2$/ms and the myocardial stimulus area consisted of 42 nodes (extension level 1). This gave an anterograde delay of about 5 ms. In Section 2.3 it was stated that physiological times for anterograde and retrograde delay were in the intervals 4-14 ms and 2-4 ms, respectively, while some articles operate with lower retrograde delay times. An anterograde delay of about 5 ms is within the given physiological range. The membrane potentials of the Purkinje node and a myocyte node close to the center of the stimulated area can be seen in Figure 6.15c.

For retrograde delay two plots are shown: Figure 6.15b shows the membrane potentials of the same Purkinje and myocyte nodes when a small part of the tissue (31 nodes out of 1331) is stimulated, of which the myocardial PMJ area is no part. In Figure 6.15d the whole tissue is stimulated at once. The first plot models a more realistic situation, and gives a delay of about 3 ms, which is within the physiological range (this value is read from the plot, and is thus not an average value over all the PMJ myocytes). The next plot is included for comparison with the realistic tree and tissue, and shows a delay of only about 1.2 ms. Stimulus on the whole tissue activates all the nodes at the same time, resulting in a great, immediate voltage difference across the PMJ, which again the Robin condition in the Purkinje problem works to diminsh. On the other hand, an activation wave coming at the PMJ area from a limited area will use some time to activate all the PMJ myocyte nodes. This will delay both the average activation time of the myocytes and the activation of the Purkinje end, as the non-activated myocytes keep the effect from the Robin condition down. From Figure 6.15b it seems that this last effect is greater than the effect on the myocardial delay, since this gives greater delay than stimulus on a limited myocardial area.

Figure 6.15a shows the action potentials at four points in space: the start node in the Purkinje tree, representing the bundle of His, a node in the middle of the tree, the end node, and the PMJ myocyte node used in the other plots just described. This figure is inspired by Figure 1.C in Deo et al. [6]. The action potentials of the Purkinje cell and the ventricular myocyte differ in duration and shape, but the model gives a gliding transition between the two, gradually shortening the action potential and creating a phase 1 notch in the Purkinje tree for cells closer to the PMJ. In this plot the differences are not very pronounced, but the tendency is still evident. The myocardium clearly affects the action potential duration in the whole tree since even the action potential at the bundle of His lasts such a short time. For such a small Purkinje tree, where the PMJ node is only three branches from the start point, this is not surprising.




**(a)** Timeseries **(b)** Retrograde

**(c)** Anterograde **(d)** Retrograde, full stimulus

**(e)** Anterograde **(f)** Retrograde

**Figure 6.15:** Action potentials. (a)-(d) are for a Purkinje tree with four branches connected at one site to a unit cube with $10 \times 10 \times 10$ elements representing the myocardium. $D_M = 0.01$ cm$^2$/ms. (a) shows the complete action potentials for different points. (b) and (c) both show retrograde propagation, but in (b) the whole tissue is stimulated (1331 nodes), while in (c) only a limited number of nodes (31), of which none are directly coupled to the Purkinje network, are stimulated. (e) and (f) are for the realistic tree and myocardium. $D_M = 0.013$ cm$^2$/ms. The points in (e) are Point 1 (black) (below Point 14) and Point 4 (gray) shown in Figure 6.16c. $D_{PMJ} = 0.06$ cm$^3$/ms, $K_1 = 3$, $K_2 = 2.945$.

### Robin Conditions and Stimulus Currents in the Realistic Case

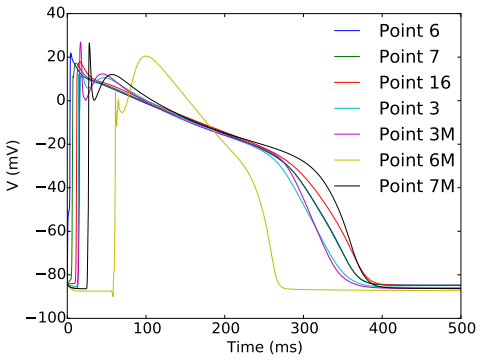
The parameter set $K_1 = 3, K_2 = 2.945$ and $D_{PMJ} = 0.06$ cm$^3$/ms, with $D_P = 0.106$ cm$^2$/ms and $D_M = 0.013$ cm$^2$/ms also worked well for the realistic case, giving quite realistic activation times for the Purkinje network and the myocardium of 27.8 ms and 65.9 ms, respectively. The activation times of the two tissues are shown in isochrone maps in Figure 6.16. An isochrone map shows when the tissue was activated. Activation is here counted as the time with steepest upstroke. The average number of myocyte nodes per PMJ was 11.

Figure 6.16d shows the activation times for the Purkinje tree. The activation starts in the bundle of His at the top of the tree, and travels evenly downwards in the left and right trees, activating the left apex last at 27.8 ms after His stimulation.

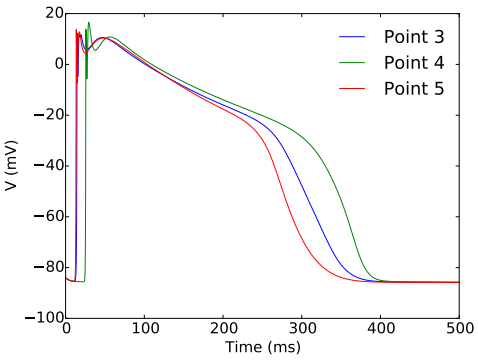
The activation times in the myocardium are shown in Figures 6.16e and 6.16f, which emphasize the left and right ventricles, respectively. The first sites to activate are those near the Purkinje ends closest to the bundle of His. In the left ventricle these points are on the septum wall separating the two ventricles, close to the top. In the right ventricle the septum is also soon to activate, as well as an area further out on the right side of the front of the heart. This last site is the first place where the activation front arrives at the epicardium (which is the outer part of the heart muscle) and breaks through the surface. This is in accordance with physiological observations, where the first epicardial breakthrough occurs in the right ventricular anterior-paraseptal region, as reported by Ramanathan et al. [22].

As the activation front continues to travel down the Purkinje tree more myocardial sites are activated, and several activation fronts in the tissue activates more tissue. The main activation pattern is apex-to-base, which also fits well with observed behaviour. This is particularly true for the posterior heart and the sides. The outer side bases of both the left and right ventricles are the last to activate, at 65.9 ms.

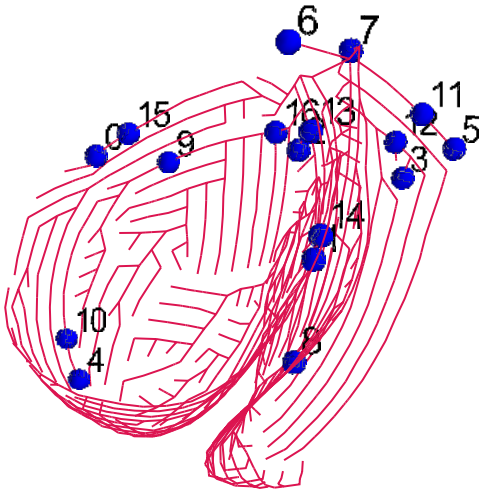
Anterograde and retrograde delay times, as well as action potentials in various parts of the Purkinje tree were studied. This was done in a manner similar to that used for the test case wirth a unit cube. Six PMJs out of the total of 109 were chosen for the study of delay times. For these PMJs, the action potentials in the Purkinje nodes and the same nodes in the myocardium were traced. The chosen points can be seen in Figure 6.16c, where they are labeled 0-5 (point 1 is below point 14, and point 2 is below and between points 13 and 16, the rest should be visible). From simulations, these myocardial sites were seen to be activated through the PMJs and not through the tissue. In addition the average delay times for all 109 PMJs



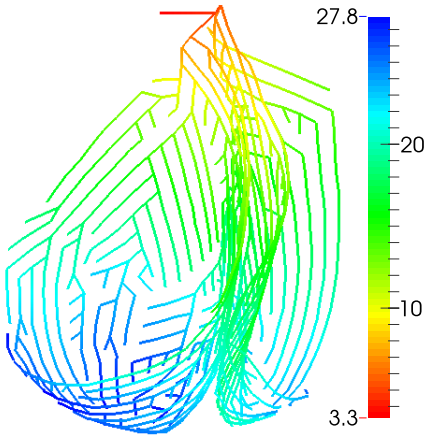

(a) Tree and myocardium



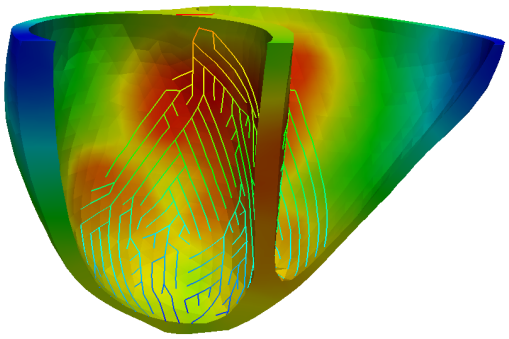

(b) PMJs



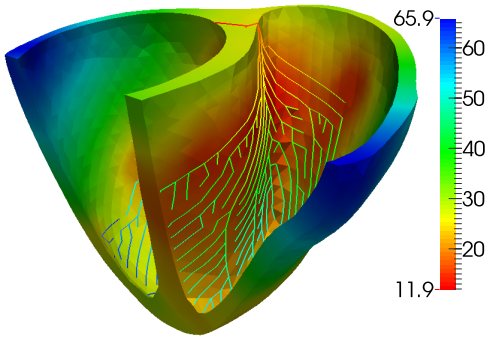

(c) Timeseries points




(d) Activation times



(e) Activation times




(f) Activation times

**Figure 6.16:** Normal activation of the Purkinje tree and myocardium. $D_{PMJ} = 0.06$ cm$^3$/ms, $D_M = 0.013$ cm$^2$/ms, $K_1 = 3$, $K_2 = 2.945$. (a) shows the timeseries of chosen points in the Purkinje tree, denoted Point x, and the myocardium, denoted Point xM. The points in the tree are shown in (c). The points in the myocardium are further described in the text. (b) shows the timeseries at chosen PMJs in the tree. (d)-(f) are isochrone maps showing the activation times of the tree and the ventricles (left and right ventricles to the left and right, respectively). (e) and (f) have the same colorscale. All colorscales have units ms.

were found. The action potentials for the Purkinje and myocardial nodes at the six PMJs showed two types of behaviour for anterograde propagation. Two representative PMJs are shown in Figure 6.15e, where the value of the membrane potential in one myocyte shoots markedly higher than the other. The delay time is significantly lower than for the unit cube, 1.63 ms on average for all the PMJs, but at least there are evident delays.

Starting a stimulus in a limited area of the myocardium and looking at the average retrograde delay makes no sense for the full Purkinje tree. The activation would then enter the Purkinje tree, and from there new myocardial sites would be activated, giving a mixture of anterograde and retrograde propagation at the PMJs. For stimulus on the whole myocardium, the membrane potentials at the six traced PMJs all showed similar behaviour. The results for one of the PMJs are shown in Figure 6.15f. This behaviour is pretty similar to the corresponding result for the unit cube. The average retrograde delay was 1.11 ms.

To see how the complete action potentials change throughout the Purkinje tree, and how they compare to results from the myocardium, action potentials were traced in all 17 points shown in Figure 6.16c, the six nodes in the myocardium corresponding to points 0-5 in the Purkinje tree, and four more points in the myocardium. Some of the results are shown in Figures 6.16a and 6.16b.

In Figure 6.16a, the bundle of His is represented by Point 6 (blue), which is the first node in the tree. This action potential looks like the action potential of a Purkinje cell. The next point, number 7 (green), is also high up in the tree, representing the right bundle branch. The upstroke of this second graph happens a little later than the upstroke of the first graph, but they both come together towards the end. This means that, although this action potential still looks like that of a Purkinje cell, it is a little influenced by the myocardium, which has shortened its action potential duration slightly, and also given it a barely visible sort of phase 1 notch. Further down the left tree is Point 16, which is nowhere near any PMJs and therefore not much influenced by the myocardium. The graph looks much the same as the first two. Shifting it along the time axis according to its later activation, the action potential duration seems to be approximately the same.

Next one PMJ is chosen, number three (cyan). The reason for this will soon be explained. This action potential clearly lasts shorter than the three before, and also has an evident phase 1 notch. The result at the corresponding myocardial node is labeled “Point 3M”, and although these graphs are very similar, there are visible differences. The last two points are from the myocardium. Point 6M (yellow) is found at the outer left top of the left ventricle, in the area that is activated the last, while Point 7M (black) is

found at the inner apex of the left ventricle. The action potential of Point 6M looks like that of a ventricular myocyte (a reduced upstroke amplitude for tissue compared to a single cell is in accordance with results from Decker et al. [15]). However, Point 7M exercise an action potential duration close to Point 3M. The reason for this is its proximity to (it might even be a part of) a PMJ area, as these are quite dense in both apexes. This was confirmed by the other two myocardial points, one of which was close to another PMJ area, and showed the same behaviour as Point 7M, while the other was far away from any PMJs (though not as far away as Point 6M), and had the same behaviour as Point 6M.

In Figure 6.16b the action potentials in three PMJ nodes in the tree are shown. They exhibit a marked difference in action potential duration. Six PMJs were traced. The three that are not shown were close to that of Point 3. The graph closest to an average was Point 3, which is why this point was used in Figure 6.16a.

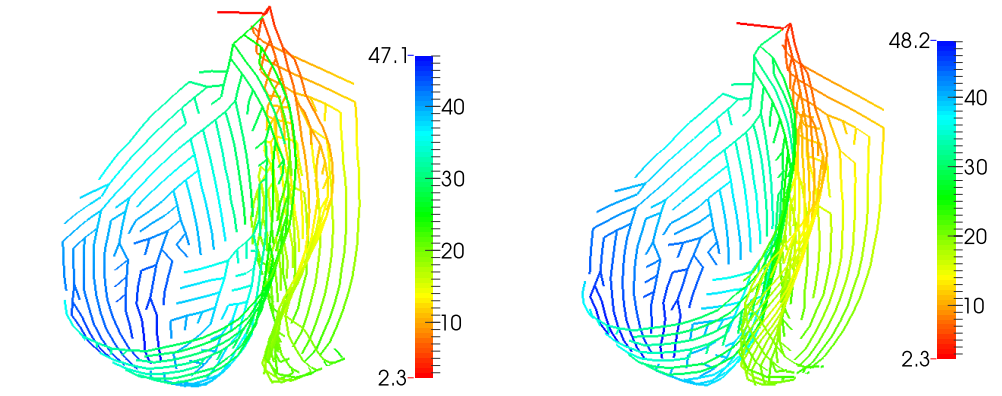
## 6.3 Left Bundle Branch Block

Left bundle branch block (LBBB) is when the left bundle branch in the Purkinje system is blocked, so that no electrical signal can pass. This leads to asymmetric activation of the ventricles, as the signal propagates normally through the right ventricle, while the left ventricle is only activated from transmural crossing of the right ventricle signal. Here LBBB is simulated simply by setting the diffusivity coefficient, which is denoted `D` in the program, in a node in the first branch in the left Purkinje tree equal to zero. The diffusivity in the other nodes have the same value (0.106 $cm^2/ms$) as before. This is done in the following code snippet:

```
V0 = FunctionSpace(mesh, 'DG', 0)
D = Function(V0)
D.vector()[:] = 0.106
D.vector()[27] = 0.0
```

The resulting signal propagation is shown as isochrone maps of the Purkinje tree and the myocardium and can be seen in Figures 6.17a, 6.17c and 6.17d. In the given order they show the Purkinje tree, the left ventricle and the right ventricle. The figures are comparable, still in the listed order, to Figures 6.16d, 6.16e and 6.16f. Comparing the plots for the Purkinje tree and the left ventricle it is clear that the activation patterns of the normal case and the LBBB case differ significantly.

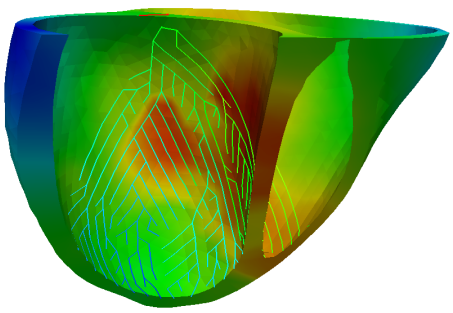
Looking at the plot of the Purkinje tree it is evident that the left ventricle



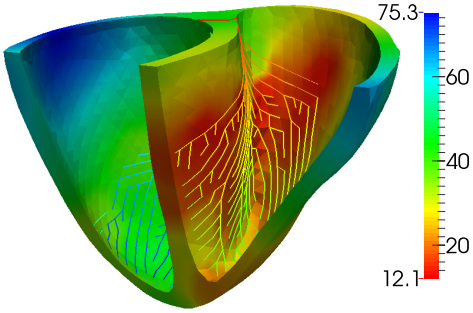

(a) $D_M = 0.013$



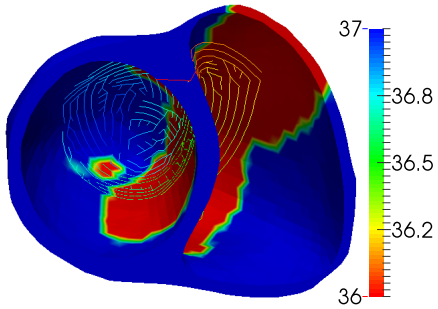

(b) $D_M = 0.01$



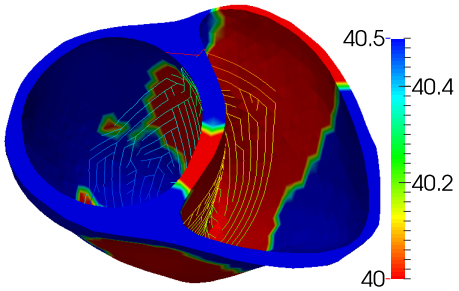
(c) $D_M = 0.013$




(d) $D_M = 0.013$




(e) $D_M = 0.01$




(f) $D_M = 0.01$

**Figure 6.17:** Isochrone maps for left bundle branch block. All colorscales have units ms. $D_P = 0.01$ cm$^2$/ms, $D_{PMJ} = 0.06$ cm$^3$/ms, $K_1 = 3$, $K_2 = 2.945$. (a), (c) and (d) have $D_M = 0.013$ cm$^2$/ms, while (b), (e) and (f) have $D_M = 0.01$ cm$^2$/ms. (c) and (d) have the same colorscale.

is activated later than the right ventricle. The part of the left ventricle that activates the last activates at 47.1 ms, while the in the normal case the last part activated at 27.8 ms. All times are relative to His stimulation. The colors for the normal case and the LBBB case are not directly comparable since the scales are not the same. This also holds for the Purkinje plots and the myocardial plots.

This is what happens: The signal starts out at the bundle of His as normal. When it reaches the first junction, that splits the network into a left and a right tree, the signal propagates onward into the right tree, while passage into the left tree is blocked. In the right tree the signal propagates as in the normal case and travels into the myocardial tissue. The first stimulation of the myocardium happens at 12.1 ms, while for the normal activation this happens at 11.9 ms. This means that during normal activation the first endocardial breakthrough site is in the left ventricle.

Among the first endocardial breakthrough sites are some sites on the upper part of the septum wall separating the left and right ventricles. From Figure 6.17c one can see that once the septum begins to activate in the right ventricle, the signal travels across the septum into the left ventricle. There is no marked delay in the activation of the septum on the side of the right ventricle as compared to the left ventricle. Now that part of the left ventricular tissue is activated, while the left Purkinje tree is not, the signal travels from the tissue into the tree. That the activation sequence is indeed this way and not the other way around can be seen from the isochrone maps. The tissue and the tree on the septum wall are both activated at around 18 ms, but while there is no visible delay between the right and left activation of the myocardium at the septum wall, there is an obvious delay between the corresponding places in the Purkinje tree (yellow in the right ventricle, green in the left ventricle). Here the delay times at the PMJs make themselves known, even though they are smaller than wanted.

From this point on the signal in the left tree spreads anterogradely towards the left apex and retrogradely towards the left bundle branch. The tissue in the left ventricle that is normally activated from the Purkinje tree is now some places activated by the tree, some places through the myocardial tissue, and some places by a combination of the two. Where the Purkinje tree is the main activation source, the myocardial activation wave is not far away, as there are no distinct places that were activated earlier than the surrounding tissue. From this it seems that the Purkinje network does not play an important role in activating the left ventricle in this particular case, as it does not manage to overtake the myocardial propagation. This will of course depend on the conductivities in both the Purkinje tree and the myocardial tissue.

The total activation time for the myocardium is 75.3 ms compared to 65.9

ms for the normal case. In the normal case the last two sites to be activated are on the top of the left and right ventricle, as can be seen in Figure 6.16e. This also holds for the LBBB case, but while the top of the right ventricle activates at about the same time as before (if there is any variation, it is so small it cannot be seen in the figures), the top of the right ventricle now activates almost 10 ms later.

The LBBB case was repeated with a slightly lower diffusion coefficient in the myocardium, 0.01 $cm^2/ms$ instead of 0.013 $cm^2/ms$, to see if the Purkinje tree in the left ventricle would then activate more of the myocardium. In the simulation it is possible to see that it does, but there is such a short difference in time between the activation from the Purkinje tree to the arrival of the activation front in the myocardium that with a scale covering the activation of the whole ventricles it is difficult to see. To make it very clear that the signal can enter the Purkinje tree through the myocardium, then move fast through the tree and activate a different part of the myocardium before the activation wave in the myocardial tissue gets to that part, the time scale was locked close to the activation time of such sites, see Figures 6.17e and 6.17f. For greater differences in the conductivities in the Purkinje network and the myocardium it seems probable that this effect will be stronger, and the Purkinje system will play a more prominent role in activating the left ventricle. Other than this feature, this LBBB case did not contribute anything new compared to the first LBBB case, and the isochrone map for the whole time span of the myocardium is omitted. The isochrone map for the Purkinje tree is shown in Figure 6.17b. The total activation times for the tree and the myocardium were 48.2 ms and 89.4 ms, respectively.

# Chapter 7

# Discussion and Concluding Remarks

A model simulating the electrical activity in the ventricles of the heart has been implemented and applied to a number of test cases. Important aspects of the model and its implementation are the representation of the ventricles and the electrical activity through meshes, cell models, tissue model, numerical methods and resolutions. Limitations of these, their importance and possible remedies are discussed below. Finally the model is evaluated in its entirety with respect to how well it represents the electrical activity in the ventricles, and what can be learned from it.

The electrical signal in the ventricles is modeled, therefore all parts conducting electrical signals must be represented. These parts are the Purkinje network and the ventricular muscle tissue. Between these two systems there is another type of cells called transitional cells, which are discussed further in Section 7.3. The data on which the representation of the Purkinje network and the myocardium are based, are based on MR images from a living human.

## 7.1 The Purkinje Network

The Purkinje system is a tree-like structure going from the SA node, branching into each of the ventricles. Purkinje fibers are very thin, which makes it challenging to obtain information about the Purkinje network through experiments. Because of this, much is still unknown, and also, interindividual structural differences seem to be large, but some characteristics seem to hold generally. According to Boyle et al. [3], the network has two distinct fascicles (main braches) in the right ventricle and three in the left ventricle. Phys-

ical observations cited in the same paper suggest that the network is quite dense. Most of the network lies on the endocardial surface, but it also has free-running components. The ends of the Purkinje network are connected to the myocardial tissue at Purkinje-myocardial junctions (PMJs). This will be further discussed in Section 7.3.

The structure of the Purkinje network influences the activation pattern and thus the activation times of the ventricular muscle tissue. It is unclear how important the exact number of fascicles in each ventricle is, but the network seems to branch more in the left ventricle than in the right. Given the larger muscle mass of the left ventricle compared to the right, a slightly larger network in the left ventricle makes sense.

A dense network means that there are many PMJs. Since this is where the electrical signal is transferred to the myocardium, the PMJ number and density plays an important role in the activation of the ventricles. In their article Behradfar et al. [2] vary the number of PMJs, using 74, 105, 166, 244, 325, 451 and 516 PMJs. They found that up to a limit, the PMJ density increases the role of the Purkinje system in activating the ventricles. For 74 PMJs, the total activation time of the ventricles was 66 ms, while for 516 PMJs this decreased to 58 ms. Increased PMJ density also increases the rate of PMJ propagation success, especially in the anterograde direction. This will be further discussed in Section 7.3.

The importance of free-running components of the Purkinje network is not yet known, but because this allows for more fiber directions than the endocardial surface this might increase the sensitivity of the ventricles to external shock (for example during defibrillation), as the activation from external stimuli depends on the orientation of the fibers with respect to the external electrical field. Free-running components might therefore introduce new activation sites. Another possible effect of free-running components is that if they are at the ends of the Purkinje network, they may change the angle of the Purkinje fiber with respect to the endocardial tissue.

The Purkinje network topology used in this thesis is a simplification of the physiological system. It is based on point data for the endocardial surface, which are based on measurements on a living human, and the program files described in Section 5.1.6, building a tree structure from these points. The branches of the Purkinje tree are represented in 1D. The complete tree consists of 839 branches, 389 in the right ventricle and 450 in the left, lying on the endocardial surface. Communication with the myocardium only takes place at the PMJs.

Purkinje fibers are represented as 1D structures. Since the fibres are very thin, 17.5 $\mu$m in radius in the PRd model [17], and are only coupled in one dimension (Decker et al. [15]), this is a reasonable approximation adopted

in several articles (for example Deo et al. [6] and Boyle et al. [3]).

The tree used does not exhibit the arborization pattern described by Boyle et al. [3] with two distinct fascicles in the right ventricle and three in the left ventricle. If this proves to be a severe drawback of the model, it can be remedied by manually enforcing such a structure, as done by Boyle et al. This might require substantial rewriting of the routine building the tree, but should not be too hard with that code as basis. The tree in the left ventricle has more branches than the right (450 to 389), so this characteristic of a slightly more developed tree in the left ventricle is well represented.

In this thesis, the number of ends in the Purkinje tree is 157, but because this lead to overlapping PMJ areas, the number of PMJs was reduced to 109. It is reasonable to assume that both 157 and 109 are much lower than the number of PMJs in any real heart. Given the results of Behradfar et al. [2], it is likely that this low number will yield the Purkinje tree a smaller activation role than it really has, increasing the activation times of the ventricles. With 109 PMJs a total activation time of the ventricles of 65.9 ms was achieved, which is in good accordance with the results of Behradfar et al. for 74 PMJs. However, other parameters play an important role too. The parameters for this particular case are given in Section 6.2.2. If the tree proves to have too low density, the tree-building routine must be given a more densely populated endocardial surface. The limitation here is thus the mesh data, not the tree-building routine.

The Purkinje tree in this thesis is built from point data, so given points that are not on the endocardial surface, a free-running component requires no change of code and is trivial. The limitation comes from the data that the tree is built from. Since external stimuli was not enforced in this thesis, and fiber directions not incorporated, the lack of free-running components should be of no importance.

It is reasonable to assume that modifications of the geometry of the Purkinje system would alter activation patterns. Here the relevant traits are specific characteristics of the structure of the tree and the density of branches, in particular PMJ density. However, despite the lack of correct number of fascicles in the ventricles and a quite limited number of PMJs, the overall behaviour of the Purkinje tree used seems to be correct, as it is able to reproduce important features such as earliest epicardial breakthrough site (in the right anterior-paraseptal region), a general pattern of apex-to-base activation and a reasonable total activation time of the network. The conclusion is then that the Purkinje tree seems to work reasonably well, and if need be, it is possible to give the tree a more correct structure and given the data points, to build a denser tree and incorporate free-running components.

The PRd cell model is used for a Purkinje cell. With its 38 ODEs this is

a very detailed model. However, the model is fitted to canine experimental data, while in this thesis the model is for a human heart. Data from mammals are often mixed and used without considerations for the specific species because they are considered to be "similiar enough". Therefore it should not be a major obstacle that the PRd model is a canine model rather than human, but this will depend on exactly what is under investigation. To the author's knowledge, there is no Purkinje cell model for humans yet.

The monodomain equation is used as tissue model for both the Purkinje network and the myocardial tissue. When coupled with a cell model it models the transmembrane potential $V = V_i - V_e$. As described in Section 3.2.1 it is a simplification of the more general bidomain equations obtained by assuming equal anisotropy rates for the intracellular and extracellular space. Since the Purkinje branches are modeled as 1D strands, they can be viewed as only consisting of intracellular space (in 1D), so the signal always moves in the fiber direction in the intracellular space. This way, the monodomain model with isotropy fully represents the signal propagation in the Purkinje network.

## 7.2 The Myocardial Tissue

The focus of this thesis has been on the Purkinje network and the coupling between the Purkinje network and the myocardial tissue. To model the details of myocardial activity has not been a main priority, therefore several simplifications affecting myocardial signal propagation have been used. This way, more time could be used on other topics, also in the way that a more detailed myocardial model would have greatly increased computation times, making adjustments of model parameters a very slow task.

The myocardial mesh is built from points based on data from a patient, so the myocardium has a realistic shape. As mentioned in the previous section, the monodomain model is used as tissue model for the myocardial tissue. This is not a very serious limitation for what is being studied here, but if need be, the monodomain equation can be extended to the bidomain equations relatively easy in FEniCS.

An additional simplification made in this thesis is to assume isotropy, so the tissue conductivity is the same in all directions. The muscles of the myocardium are organized in a swirling pattern, which makes the heart contract and pump blood efficiently, so isotropy is not a realistic approximation. This simplification is made because the model does not incorporate fiber directions, so anisotropy would not be meaningful. If fiber directions are included, anisotropy can easily be implemented. The lack of fiber directions is the greatest limitation of the model in this thesis, and these must be included

to model realistic activation in the myocardium.

The ventricular muscle tissue consists of several layers. From the inside out one finds the endocardium, the myocardium and the epicardium. In this thesis the endocardium, the myocardium and the epicardium are all represented as the myocardium, as is also the case in the articles by Vigmond and Clements [26], Deo et al. [6] and [7], Boyle et al. [3] and Behradfar et al. [2]. Depending on what is being studied this can be a serious limitation but should not be so here.

The Decker et al. cell model is used to model the cells in the myocardium. As the PRd cell model this is a highly detailed model, with 46 ODEs. This model is for a canine epicardial myocyte not for a human ventricular myocyte. If detailed myocardial activation is in focus other models exist, also for humans. Since human data exist for ventricular cells it is also possible to adjust model parameters to better reproduce observational data.

## 7.3 Coupling of the Purkinje Network and the Myocardial Tissue

Communication between the Purkinje network and the myocardium only happens where the ends of the Purkinje network meet the myocardial tissue. The coupling sites are called Purkinje-myocardial junctions (PMJs), and the communication happens via transitional cells, as described in Section 2.3. One Purkinje cell is indirectly coupled to several ventricular myocytes. The transitional cells work as a high resistance barrier between the Purkinje network and the myocardial tissue. This barrier gives delay times at the PMJs. Retrograde propagation is highly favored over anterograde propagation. The angle between the ends of the Purkinje network, ending in transitional cells, and the fibre direction of the myocardial tissue influence the probability of propagation success or failure.

The model in this thesis only allows for communication between the Purkinje tree and the myocardium at the PMJs, so this characteristic is well represented. The Purkinje tree and the myocardium are represented as two separate meshes, so communication between them must be implemented explicitly. Aspects that need to be considered are which areas in the Purkinje mesh that should be coupled to which areas in the myocardial mesh, and how this coupling should be done. Several alternatives were tested for each of these aspects.

In the Purkinje tree the end nodes were used as PMJ areas, so there is one Purkinje node per PMJ. It is not as obvious what should be chosen as PMJ

areas in the myocardium. In the meshes used, the end nodes in the Purkinje tree are also nodes in the myocardial mesh, so it seemed natural that these nodes should at least be part of the myocardial PMJ areas. Alternatives that were tested were a single myocyte node per PMJ area, one facet including the original node, one element including the mentioned facet, all facets including the original node, all elements including the mentioned facets, and then more elements building from the two element-alternatives mentioned, where "layers" of adjacent elements were added in a hemispherical way as many times as wanted.

The ends of the Purkinje tree are assumed to go a bit into the myocardium. Exacty how far in they go is not yet known, different studies operate with different distances, but Boyle et al. [3] let the ends go up to 20 % into the thickness of the endocardial tissue. This feature is not at all represented when the myocardial PMJ areas are single nodes or facets, but when they are elements this is somewhat represented. The PMJ areas have room for improvement regarding shape, but at the present the mesh resolution is probably a greater limitation. Given a finer mesh the present solution will better represent reality, and if need be it should not be very difficult to improve these areas.

The angle between the ends of the Purkinje network and the myocardial fiber directions influence the propagation success, a lower angle increasing the success rate. Especially anterograde propagation success is affected by the angles as this is already the least favored propagation direction. Boyle et al. [3] found that in their model about 75 % of the PMJs remained quiescent during anterograde propagation even though the complete Purkinje system was activated. Since fiber directions are not incorporated into this model at the present, the feature of angles has not been considered, but important activation characteristics are still reproduced (see Section 5.2.1).

As mentioned in Section 5.2.1, PMJ density also affects propagation success. This is probably because stimulus on a larger area increase the probability of successful activation and if the PMJ density is high, several such areas combined can create a larger, almost continuous, activation area. This effect should be present in the model but is probably not very visible, PMJ areas being as large as they are.

The methods used for coupling the Purkinje network and the myocardial tissue were Dirichlet conditions for the myocardium for a one-way coupling, enabling propagation from the Purkinje tree to the myocardium, and for a two-way coupling Dirichlet conditions for the myocardium and Neumann conditions for the Purkinje tree, Robin conditions for both systems, and Robin conditions for the Purkinje network and stimulus currents for the myocardium. The values at the ends of the Purkinje tree were used in the

Dirichlet conditions and the flux out of these ends were used for the stimulus currents. The flux through PMJ facets in the myocardium were used as Neumann conditions for the Purkinje tree.

For the one-way coupling with Dirichlet conditions for the myocardium, only one myocyte node was used per PMJ area. This gave anterograde propagation and worked well for a one-sided connection where PMJ delay times were not considered. For the two-way coupling with Dirichlet conditions for the myocardium and Neumann conditions for the Purkinje tree, a myocardial facet was needed for the Neumann condition for each PMJ, therefore the myocyte area was extended to a facet. This PMJ area better represents that one Purkinje cell is indirectly coupled to several ventricular myocytes. To achieve propagation both ways at the PMJs for this coupling method, some adjustments were needed. This method is supposed to give continuity of the membrane potential and conservation of current at each PMJ. However, non-physical scaling of parameters in the implementation leads to this no longer being completely maintained.

Transitional cells are not included explicitly in this model or any other model to date to the author's knowledge. Instead parameters and equations are tuned to reproduce physiologically valid PMJ delay times both in the anterograde and retrograde directions. Delay times at the PMJs presuppose discontinuities between the Purkinje network and the myocardial tissue. In reality there is, of course, continuity, but this is between the Purkinje tree and the transitional cells and between the transitional cells and the myocardium. The membrane potential is not continuous between the Purkinje tree and the myocardium. The two methods described above using Dirichlet conditions strive for continuity of membrane potentials at the PMJs, so they cannot give marked delay times. For a model to be able to reproduce such delay times the characteristics of the transitional cells must somehow be represented in the coupling.

The expressions for the Robin conditions include a term representing the conductance between the Purkinje tree and the myocardium: $\sigma_{PMJ}$. This can be seen as a representation of the transitional cells between the Purkinje cells and the ventricular myocytes. The conductance at the PMJs should be between the conductances in the Purkinje and myocardial tissues. If the PMJ conductance is too high it represents too high electrotonic loading from the myocardium so the potential in the Purkinje ends will die out, and if it is too low, the signal dies out in the myocardium. In the implementation, diffusivities, rather than conductivities, are used for the Purkinje tree and the myocardium. At the PMJs it is not exactly the diffusivity that is used, so the values are not directly comparable, and the expression for the stimulus currents is not completely correct, which may account, at least in part, for

the scaling necessary to get bidirectional propagation.

Robin conditions for both the Purkinje and the myocardial problems were combined with facets as myocardial PMJ areas, while Robin conditions for the Purkinje tree and stimulus currents for the myocardium were tested with all mentioned areas except for a single node (some areas on a unit cube, some on the realistic meshes). Both methods needed adjustments to give bidirectional propagation. The last approach was able to give noticeable PMJ delay times and higher anterograde than retrograde delay times, but not quite in the range assumed to be physiologically reasonable.

A possible source of trouble for the coupling is that the myocardial PMJ areas do not hold enough nodes. More nodes favor anterograde propagation to a greater extent but in a mesh as coarse as the one used, all areas that consist of several nodes are larger than what is physiologically reasonable. Even a single facet per PMJ leads to overlapping areas, and enlarging the area even more without special treatment of each single area will also lead to either even more overlap or weirdly shaped areas. This is a drawback with all the myocardial PMJ areas that consist of several nodes and is a limitation due to the resolution of the myocardial mesh. For coarse meshes there must be a balance between the size of the PMJ areas and the number of nodes in them. Too few myocardial PMJ nodes might account for the adjustments necessary to achieve bidirectional propagation. These adjustments favor PMJ propagation, so if they were rendered unnecessary one could hope to get delay times closer to the physiological range.

It might be that other myocardial PMJ areas for Robin conditions for both problems would have been able to give PMJ delay times, but this was not tested. Still, it is reasonable to assume that this would not have given as pronounced delay times as stimulus currents for the myocardium since stimulus currents give a lower slope in the beginning of the depolarization phase of an action potential. This lower slope occurring in the anterograde direction may also make the delay time in this direction greater than in the retrograde direction, in accordance with observations. This last feature is then a phenomenological property of the model, as in reality one must assume that this characteristic is only caused by the greater mass of the myocardium compared to the Purkinje tree and the one dimensionality of the Purkinje tree.

To sum up, two main aspects were found to be important when implementing the Purkinje-myocardial coupling. These were the choice of myocardial PMJ areas and the coupling method. All the two-way methods needed parameter adjustments to give bidirectional propagation. The greatest limitation of the coupling is probably due to the resolution of the myocardial mesh, which restricts the choice of PMJ areas.

## 7.4 Resolution

Sundnes et al. [24, p.17] state that for realistic heart cell models, resolving the fast dynamics of the upstroke properly requires a spatial discretization of about 0.2 mm and a temporal resolution of about 0.1 ms. In accordance with this a temporal resolution of 0.1 ms was used in this thesis, which seemed to work well, giving propagation through the Purkinje and myocardial meshes.

The spatial resolution of the Purkinje network can be controlled in the file `call_tree_msh.py` by the parameter `ds`, which is measured in cm. Choosing `ds` = 0.05 yields most element lengths equal to 0.5 mm, but some can be up to 1.0 mm. This is considerably higher than the value of 0.2 mm given by Sundnes et al. but gave reasonable signal conduction, as could be seen from the simulations. The present Purkinje tree has 6357 nodes. If higher resolution is needed this can easily be fixed by giving `ds` a lower value.

The size of an element in FEniCS is characterized either by the diameter of the sphere circumscribing the element or the diameter of the inscribed sphere. For a mesh, the maximum and minimum values of both these measures can be given by built-in functions. For the myocardial mesh used here, the diameters of the circumscribed circles are in the range 1.559-15.747 mm, while the diameters of the inscribed circles are in the range 0.202-3.718 mm. Except for the lowest value of the inscribed circles these values are much higher than the desired value of 0.2 mm. This makes it difficult to define good myocardial PMJ areas, as discussed earlier, and to get propagation in the myocardial mesh one must use unrealistic conductivity values (diffusion coefficients in the program). The mesh used for the myocardium consists of 3584 nodes.

In FEniCS it is easy to refine meshes. The reason this was not done in this thesis is that the number of nodes increases very fast, and with it the simulation times. The total number of nodes in this problem was 9941. This is not much considering the problem, in the simulations of Deo et al. [6] they use about 550 000 nodes, which gives element edge lengths of about 0.3 mm, but special computers are required for such heavy calculations. The computer used to run the simulations in this thesis was a MacBook Pro with an Intel Core 2 Duo processor and 4 GB RAM. A virtual machine with ubuntu 64 bit was run on this machine, with access to 2560 MB RAM.

## 7.5 Numerical Methods

A combination of numerical methods were applied to solve the monodomain equation for the Purkinje system and the myocardium. The error due to time discretization is proportional to the time step $\Delta t$ raised to the power of

the order of the method, and the same holds for spatial methods only with the spatial step $\Delta x$. This means that the orders of the methods in time and space need not be the same. The accuracy of the solution is limited by the lowest accuracy of all the methods. Better accuracy can be achieved either by increasing the order of the methods or by decreasing the discretization steps. The methods that were applied were operator splitting and the finite difference method in time, and the finite element method in space.

For operator splitting Godunov splitting was used. This method is first-order in time, so the error is proportional to $\Delta t$. The time step used is $\Delta t = 0.1$ ms. Since the accuracy of the solution is limited by the method with the lowest accuracy, a first-order method was also chosen for the finite difference method in time. One can then choose between the forward and backward Euler methods, but since the forward Euler method can be unstable, depending on the time step, the backward Euler method was used.

In the finite element method the simplest basis functions that give a continuous solution are linear functions. While the original problem demands that the solution is twice differentiable, the weak problem only requires that the squares of the derivatives are integrable (Lines [18, p.41]), which is fulfilled for linear functions. Linear functions give second order accuracy, so the error is proportional so $\Delta x^2$. As described in Section 7.4 the spatial discretization used for the myocardium in this thesis is very coarse.

The orders of the methods are based on Lines [18, Chapters 4 and 6]. There a first order method is used in time and linear elements in space for a bidomain model of the ventricles embedded in a torso. Lines found that a spatial resolution of 0.2 mm and a time resolution of 0.125 ms are reasonable for that problem. In this thesis a time step of the same magnitude as Lines found reasonable is used, while the spatial resolution is coarser, which is fine since other conductivity values are used.

## 7.6 Concluding Remarks

The goal of this thesis was to model the electrical activity in the ventricles with particular focus on the Purkinje-myocardial coupling. Several models that are able to simulate the activation of the ventricles through the Purkinje tree and the myocardial tissue have been implemented, where the Purkinje-myocardial junctions (PMJs) are represented in various ways.

The representations of the PMJs have been attacked from a physiological, mathematical/numerical and implementational point of view. Through literature studies the PMJ physiology at the cell-level and its implications for organ-level behaviour have become known. The representation of PMJs

in earlier modeling studies have been mathematically described (Robin conditions for the Purkinje problem and stimulus currents for the myocardial problem), as well as other known coupling methods. Several methods have been implemented, particular care being invested in achieving reasonable myocardial PMJ areas and reproducing known activation characteristics and special PMJ effects.

Major drawbacks of the models are that fiber directions in the myocardium are not represented and the resolution of the myocardial mesh is too coarse, which can be a problem in itself, but also causes problems for the choice of myocardial PMJ areas. It can also be desirable with a denser Purkinje network and to extend the monodomain model to the bidomain model. As mentioned earlier in this chapter, it is relatively easy to extend the monodomain equations to the bidomain model and to include myocardial fiber directions. The reason this was not done in this thesis was that the focus was on PMJ coupling, and these changes would substantially increase computation times. The same is true for refinement of the myocardial mesh, which would have allowed realistic conductivity values. A denser Purkinje tree is probably the greatest challenge but should not be unsurmountable. Also, this is not really a drawback of the models but a result of limited observational data.

Despite the drawbacks of the models, they are able to activate the ventricles in a convincing manner. The model with Robin conditions for the Purkinje problem and stimulus currents for the myocardial problem is able to reproduce important features such as bidirectional propagation at the PMJs, earliest epicardial breakthrough site in the right anterior-paraseptal region, a general pattern of apex-to-base activation, PMJ delay times in both directions, favoring retrograde propagation, and reasonable total activation times for the Purkinje network and the myocardial tissue.

The purpose behind the implemented models is that they should be able to simulate realistic cases of ventricular arrhythmia. In this thesis time did not allow for investigations of arrhythmia beyond the simple study of left bundle branch block presented in Section 6.3, which was done with the model with Robin conditions for the Purkinje problem and stimulus currents for the myocardial problem. For more realistic simulations, the models and their implementations should be further validated, and the major drawbacks mentioned above must be addressed.

# Bibliography


[1] David A.Rawling and Ronald W.Joyner. Characteristics of Junctional Regions Between Purkinje and Ventricular Muscle Cells of Canine Ventricular Subendocardium. *Circulation Research*, 1987;60:580-585.

[2] Elham Behradfar, Anders Nygren, and Edward J. Vigmond. The Role of Purkinje-Myocardial Coupling during Ventricular Arrhythmia: A Modeling Study. *PLoS ONE 9(2): e88000. doi:10.1371/journal.pone.0088000*, February 2014.

[3] Patrick M. Boyle, Makarand Deo, Gernot Plank, and Edward J. Vigmond. Purkinje-mediated Effects in the Response of Quiescent Ventricles to Defibrillation Shocks. *The Journal of the Biomedical Engineering Society*, October 2009, 10.1007/s10439-009-9829-4.

[4] PRd cell model code. http://rudylab.wustl.edu/research/cell/code/Cells_IonChannels.html.

[5] CellML. http://models.cellml.org/e/188/decker_2009.cellml/@@docgen.

[6] Makarand Deo, Patrick M. Boyle, Albert M. Kim, and Edward J. Vigmond. Arrhythmogenesis by Single Ectopic Beats Originating in the Purkinje System. *American Journal of Physiology - Heart and Circulatory Physiology*, October 2010, Vol.299 no.4: H1002-H1011.

[7] Makarand Deo, Patrick M. Boyle, Gernot Plank, and Edward J. Vigmond. Arrhythmogenic Mechanisms of the Purkinje System During Electric Shocks: A Modeling Study. *Heart Rhythm*, December 2009, Vol.6 issue 12, Pages 1782-1789.

[8] D.P.Zipes, A.J.Camm, M. Borggrefe, A.E.Buxton, B.Chaitman, M.Fromer, G.Gregoratos, G.Klein, A.J.Moss, R.J.Myerburg, S.G.Priori, M.A.Quinones, D.M.Roden, M.J.Silka, and C.Tracy. ACC/AHA/ESC

2006 guidelines for management of patients with ventricular arrhythmias and the prevention of sudden cardiac death: a report of the American College of Cardiology/American Heart Association Task Force and the European Society of Cardiology Committee for Practice Guidelines (Writing Committee to Develop Guidelines for Management of Patients With Ventricular Arrhythmias and the Prevention of Sudden Cardiac Death). *Circulation*, 2006;114:e385–e484.

[9] Dødsårsaksregisteret. http://www.ssb.no/dodsarsak.

[10] FEniCS. http://fenicsproject.org.

[11] Johan Hake. https://bitbucket.org/johanhake/goss.

[12] Johan Hake. https://bitbucket.org/johanhake/goss/src/e9439b6a4f87c02d462776e153ea5bfb138dc2bb/test/cpp/fhn.ode?at=master.

[13] Johan Hake. https://bitbucket.org/johanhake/gotran.

[14] James Keener and James Sneyd. *Mathematical Physiology 1: Cellular Physiology*. Springer, 2. edition, 2009.

[15] K.F.Decker, J.Heijman, J.R.Silva, T.J.Hund, and Y.Rudy. Properties and Ionic Mechanisms of Action Potential Adaptation, Restitution, and Accommodation in Canine Epicardium. *American Journal of Physiology - Heart and Circulatory Physiolog*, 2009, Vol.296, no.4, April.

[16] Richard E. Klabunde. http://cvphysiology.com/Arrhythmias/A003.htm.

[17] Pan Li and Yoram Rudy. A Model of Canine Purkinje Cell Electrophysiology and $Ca^{2+}$ Cycling: Rate Dependence, Triggered Activity, and Comparison to Ventricular Myocytes. *Circulation Research*, 2011.

[18] Glenn Terje Lines. *Simulating the Electrical Activity of the Heart. A Bidomain Model of the Ventricles Embedded in a Torso.* Unipub forlag/Akademika AS, 1999.

[19] Study material for the course INF5610 at the University of Oslo. http://www.uio.no/studier/emner/matnat/ifi/INF5610/h08/undervisningsmateriale/markov.pdf.

[20] Rahul Mehra. Global public health problem of sudden cardiac death . *Journal of Electrocardiology*, 2007, Vol.40(6), November.

[21] S. G. Priori, E. Aliot, C. Blomstrom-Lundqvist, L. Bossaert, G. Breithardt, P. Brugada, A. J. Camm, R. Cappato, S. M. Cobbe, C. Di Mario, B. J. Maron, W. J. McKenna, A. K. Pedersen, U. Ravens, P. J. Schwartz, M. Trusz-Gluza, P. Vardas, H. J. J. Wellens, and D. P. Zipes. Task Force on Sudden Cardiac Death of the European Society of Cardiology. *European Heart Journal*, 2001, Vol.22, issue 16, August.

[22] Charulatha Ramanathan, Ping Jia, Raja Ghanem, Kyungmoo Ryu, and Yoram Rudy. Activation and Repolarization of the Normal Human Heart under Complete Physiological Conditions. *PNAS, vol.103, no.16, 6309-6314*, April 2006.

[23] Rudy. Properties and ionic mechanisms of action potential adaptation, restitution, and accommodation in canine epicardium. http://models.cellml.org/e/188/decker_2009.cellml/@@docgen.

[24] Joakim Sundnes, Glenn Terje Lines, Xing Cai, Bjørn Fredrik Nielsen, Kent-Andre Mardal, and Aslak Tveito. *Computing the Electrical Activity in the Heart*. Springer, 1. edition, 2010.

[25] Jørgen Tranum-Jensen, Arthur A.M.Wilde, Jessica T.Vermeulen, and Michiel J.Janse. Morphology of Electrophysiologically Identified Junctions Between Purkinje Fibers and Ventricular Muscle in Rabbit and Pig Hearts. *Circulation Research*, 1991;69:429-437.

[26] Edward J. Vigmond and Clyde Clements. Construction of a Computer Model to Investigate Sawtooth Effects in the Purkinje System. *IEEE Transactions on Biomedical Engineering*, March 2007, Vol.54, no.12, Pages 1782-1789.