A comparison of norms for characterizing numerical solutions arising in simulations of the electrical cardiac activity

by

Ivar W. K. Framnes

MASTERTHESIS
for the degree of
MASTER OF SCIENCE

(Master i Anvendt matematikk og mekanikk)

Faculty of Mathematics and Natural Sciences
University of Oslo

Mai 2011

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

Contents

Acknowledgments

First of all I want to thank my advisors Ola Skavhaug, Glenn Terje Lines and Aslak Tveito for their comments and suggestions while I worked on this thesis. Especially thanks to Glenn for spending extra time to help me in the final stage of my work with the thesis. Further, I would like to thank Sven Haadem for giving me good advice, Damir Nedic for motivating me and my parents and brothers for all their love and care. Lastly, a special thanks to my dear Ida Solhjell for proofreading my thesis, and for encouraging me.

Chapter 1

Introduction

1.1 The Heart and Fibrillation

The heart is considered one of the most important organs. By pumping blood through the circulatory system the heart transports vital oxygen from the lungs to the rest of the body. Even if a heart failure is non-lethal it can lead to irreparable damage to other organs, such as the brain. If the brain is denied oxygen, it will be permanently damaged in just a matter of minutes. In the western part of the world, heart related illnesses are by far the most common cause of death. Gaining a greater understanding of the heart is thus of great importance.

The human heart has a mass of between 250 and 350 grams and is about the size of a fist. It has four chambers: two superior atria and two inferior ventricles. The atria are the receiving chambers, while the ventricles are the discharging chambers. A single heart beat consists of a series of electrochemical events causing an electrical wave to propagate through the cardiac tissue. This is a well-synchronized process which results in a rhythmic contraction of the cardiac muscle. However, certain pathological conditions can destabilize this electrical wave, leading to cardiac arrhythmia and causing fibrillations. These fibrillations take on different forms with varying degree of severity. A ventrical fibrillation is an uncoordinated contraction of the ventrical chambers, and makes the heart quiver. This fibrillation is lethal within minutes if not treated.

When simulating the electrical activity in the cardiac muscle, it is of great interest to see how changes to physiological parameters can cause destabilization of the heart's electrical activity. Distinguishing physiological properties leading to electrical instabilities resulting in cardiac fibrillation, is of particular interest. It is therefore essential that we are able to compute whether or not a specific simulation is fibrillatory. Observed by the human eye, this is easy to recognize, and can easily be distinguished from stable behavior. The difficulty lays doing so in a computational manner. A numerical technique for analyzing stability would enable us to run large scale series of simulations and automatically deduce whether the solution is fibrillatory or not.

The electrical activation of the cardiac muscle is a well-studied phenomenon, and offers a wide range of mathematical models describing the different electrophysiological properties of the heart. In this thesis we will implement a simple mathematical model to simulate cardiac fibrillation, and look at different ways of numerically measuring the results and exploring how to distinguish turbulent from laminar flows.

1.2 Diagnostic Models

All the infomation given in this section comes form [5]. There exist several diagnostic devices for analyzing heart conditions. The most commonly used technique is the easily recognizable electrocardiogram. This technique is in fact the oldest of the noninvasive tools, and was first published in 1887 by Augustus D. Waller. Waller held several demonstrations on his technique, many of which where on his dog, Jimmy. With Jimmy's paws submerged in buckets of saline the dog's paws acted as electrodes. As it is impossible to measure the electrical potential for a single point, Waller recorded the potential difference between Jimmy's rear and front paws. He observed that the potential difference pulsated in sync with the rhythm of Jimmy's heart beat. Later Waller presented evidence that supported his idea that this potential difference resulted from the electrical activity in cardiac muscle. Waller was the first to name the technique electrocardiogram, or ECG.

Willem Einthoven, who had been attending one of Waller's demonstrations, was intrigued by Waller's method and came up with an idea for refining it. Einthoven submerged a person's hands and left leg in the conductive saline solution. With a third electrode, he was able to make the ECG more sensitive and at the same time more robust. The ECG with three electrodes was able to measure the potential difference between each of the them, resulting in three leads instead of only one. These leads where defined as

$$\begin{aligned} I &= \phi_{LA} - \phi_{RA} \\ II &= \phi_{LL} - \phi_{LA} \\ III &= \phi_{LL} - \phi_{RA}, \end{aligned}$$

where ϕ_{LL}, ϕ_{LA} and ϕ_{RA} denoted the potential measured at the left leg, left arm and right arm, respectively. These leads were bipolar leads, in the sense that they recorded the potential difference between two points.

Viewing the body as a volume conductor, the electrical current caused by the cardiac muscle can be thought of as a dipole. An electrical dipole is a pair of closely spaced poles with opposing charge, but with equal magnitude $(-q, q)$. The dipoles generate an electrical field, causing current to flow through the conductive medium, which again is measured by the ECG. The dipole moment measures the electrical polarity of a system of charges, and can be given by

$$\mathbf{p} = q\mathbf{d},$$

where the dipole moment is $\mathbf{p}$, and $\mathbf{d}$ is a vector from the negative to the positive pole. During the activation of the cardiac tissue, the current sources can be approximated by a number of dipoles and respectable dipole moments. The sum of these dipole moments is the heart vector. This vector describes the sources of electrical current in the tissue.

Einthoven's ECG made it possible to create a projection from the heart vector onto the three leads. With this model it was possible gather a lot of information from just three leads ECG. One might think that no additional information could be obtained by adding further leads. In fact, this would have been the case if the heart truly was dipole in the frontal plane, defined

by just these three leads. However, this simplified view of the heart is not always sufficient. Especially, in the cases where the heart vector was not oriented in the frontal plane. Also, the dipole approximation could not fully reproduce the complicated electrical activity in cardiac tissue.

A group of scientist led by Wilson invented the next generation of ECG's. Since the electrical potential had to be measured relative to some reference potential, an independent reference, or zero reference, would be helpful. This zero reference should preferably be constant during the heart cycle. As no electrical charge enters or leaves the body during the heart cycle, the sum of all potential had to be zero. Wilson and his group constructed an independent reference by connecting all three of Einthoven's electrodes. These leads would approximate the potential generated by the entire body.

Wilson and his group kept Einthoven's former electrodes and added six new ones. These six electrodes were connected to the front of the chest and defined the unipolar leads, $V1 - V6$. These leads were unipolar as the potential difference recorded was measured using the independent reference. In 1938, $V1 - V6$ together with Einthoven's leads constituted the standard nine-lead ECG.

In 1942, Goldberger improved the ECG even further by including three additional leads; aVR, aVL and aVF. These were all unipolar leads each connected to the three electrodes introduced by Einthoven. This is the standard twelve-lead ECG used today, but there are still discussions on whether or not more leads should be added.

When trying to understand the underlying physiology of the ECG, it is necessary to study the electrochemical reactions that take place in the cardiac muscle. Cardiac cells are part of a class the cells called excitable cells. These cells have the ability to respond actively to electrical stimulus. Other examples of excitable cells are nerve and skeletal cells. While in resting state, excitable cells maintain an internal ionic concentration different from its surroundings. This means that the electrical charge of ions in the cell results in a potential difference across the cellular membrane. This potential difference is called transmembrane potential or simply membrane potential.

If electrical stimulus is applied to an excitable cell, it will respond according to one of two possible patterns: If the electrical stimulus is small, the membrane potential will become slightly elevated and quickly return to its resting value again. On the other hand, if the stimulus is sufficiently strong, and able to raise the transmembrane potential to some threshold level, the respond is very different. In this case, the conductive property of the cell membrane changes, resulting in a rapid flux of ions onto the cell. This causes depolarization, lifting the transmembrane potential to some peak value, which is either around zero or significantly above zero, depending on cell types inspected. After the quick depolarization phase, the membrane potential slowly is lowered to its normal resting value. This phase is called the repolarization phase and the complete process with de- and repolarization is called action potential. In many excitable cells, the repolarization phase lowers the transmembrane potential quite rapidly. However, for cardiac cells the membrane potential lingers for some time around its depolarized state. This is called the plateau phase.

Figure 1.1 shows the potential difference across a lead during a typical heart beat. The straight line segments appear when the potential difference is zero, corresponding to the intervals in the cardiac cycle when there are no source terms in the cardiac tissue. The five deflections occur during the electrical activation of the cardiac cells. Einthoven identified these five deflections as the P-wave, followed by the QRS-complex, and lastly the T-wave. The P-wave are recorded when the Atria, the smaller heart cav-

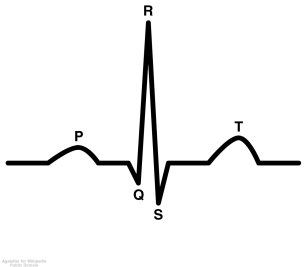

Figure 1.1: The sketched signal showing the potential variation measured over a single lead during a cardiac cycle

ities, are being depolarized. The QRS-complex, indicate the activation of the larger heart cavitis; the ventricles. The final T-wave characterizes the

repolarization of the whole cardiac muscle.

During the last decades, there has been great development in understanding and modeling biological systems. Studies of cellular and sub-cellular processes have been refined, paving the way for advanced mathematical models describing biological phenomena. Several models have been constructed characterizing the electrophysiology of the cardiac muscle, examining the heart in different scales. The concept of a heart vector is an example of such a model introduced nearly a hundred years ago. The heart vector model, however, was based on the top down approach, and did not take into account the underlying physiology.

In 1958, Hudgkin and Huxley proposed a quantitative model for wave propagation in excitable cells. The Hudgkin and Huxley model was based on detailed models of ionic currents. This greatly impacted the modeling of various biological phenomena. While Hogkin and Huxely's model only contained four ordinary differential equations with only one of which described the ionic current, their model created a basis for the development of other more sophisticated models.

In 1962, D. Noble developed the first physiological model of cardiac tissue. Since then, several more realistic models have been developed, some of which even incorporates single cell processes. Though the acute degree of detail made these models computationally strenuous, the rapid development of computational hardware and numerical techniques have helped making them viable simulators. These models have surpassed the analytical approach of the ECG, and additionally also have the ability to predict heart behavior.

1.2.1 This Thesis

In Chapter 2 we will look at some mathematical models for simulating action potential propagation in cardiac tissue. We will explore which models are best suited for reproducing fibrillatory patterns. The model will be a set of differential equations, and in Chapter 3 we will examine different numerical methods for solving these equations. Choice of solver will be determined by its accuracy and stability, for a time efficient solver. In Chapter 4, we initiate different patterns of fibrillations, based on the models introduced

in Chapter 2, and compare these patterns by using some well-known norms to distinguish between stable and non-stable fibrillations. We will refer to break-up solutions as unstable or chaotic.

Chapter 2

Mathematical Modeling of Cardiac Tissue

2.1 Bidomain and Monodomain Model

There exists a number of models describing the electrical activity in cardiac tissue. These models vary in level of detail and are dependent on the biophysiological phenomenon of interest. Modeling each cell as a separate unit before coupling them together, offers a great level of detail and precision. However, the vast number of cells makes this approach extremely numerically strenuous. The bidomain model was developed in the late 1970s and is used extensively in numerical simulations of electrical behavior in the heart [8]. These model are based on volume-averaging techniques when predicting electrical behavior in cardiac tissue. Rather than treating every cardiac cell as separate entities, they model a quantity of cells at a given point P as an average for some ball, B_P, surrounding P. These balls are scaled so that they are small compared to the domain, but large in comparison to a single cell [5].

The bidomain equations are a set of coupled partial differential equations governing the intracellular potential v_i and extracellular potential v_e. On the interior of the domain, Ω, both the extracellular and intracellular regions

of the electrical potential satisfy these conservation equations [4]:

$$\nabla(M_i \nabla v_i) = \chi I_m, \tag{2.1}$$
$$\nabla(M_e \nabla v_e) = -\chi I_m. \tag{2.2}$$

The electrical conductivity is represented by the parameters M_i and M_e corresponding to the intra- and extracellular domain, respectively. I_m is the transmembrane current density, and is given by

$$I_m = C_m \frac{\partial(v_i - v_e)}{\partial t} + I_{ion}, \tag{2.3}$$

where I_{ion} is the ionic current defined which will be discussed further in Section 2.2. The parameter χ denotes the ratio between surface area and the volume of the cell membranes, while C_m is the electrical capacitance of the tissue. The parameters M_i, M_e and χ represent the discrete structure of the tissue averaged over a scale of many cell lengths [4].

The transmembrane potential v of the cardiac muscle is defined as:

$$v = v_i - v_e \Rightarrow v_i = v + v_e.$$

This eliminates v_i for equation (2.1), and results in the standard formulations

$$\begin{aligned} &\nabla(M_i \nabla v) + \nabla \cdot ((M_e + M_i)\nabla v_e) = 0, \\ &\nabla(M_e \nabla v_e) = -\chi I_m. \end{aligned} \tag{2.4}$$

The cardiac muscle is surrounded by insulator material which is reflected in the boundary conditions [7]

$$\begin{aligned} &\mathbf{n} \cdot (M_i \nabla(v_e + v)) = 0, \quad \ltimes \cdot (M_e \nabla V_e) = 0 \quad \text{on } \partial\Omega_1, \\ &\mathbf{n} \cdot (M_i \nabla(v_e + v)) = 0, \quad v_e = v_{stim} \quad \text{on } \partial\Omega_2. \end{aligned} \tag{2.5}$$

The division of boundary $\partial\Omega$ into $\partial\Omega_2$ and $\partial\Omega_1$ refers to the sinoatrial node. The sinoatrial node is the impulse-generating tissue located on the wall of the right atrium of the cardiac muscle, and generates the normal sinus rhythm causing cardiac contraction. This is accounted for in the $\partial\Omega_2$ part of the boundary conditions.

Cardiac tissue is anisotropic, meaning that the electrical conductivity is directionally dependent. The anisotropy property is determined by molecular, cellular and histological determinants [10]. This property is represented by the conductivity tensors M_i and M_e, which for three dimensions are given as:

$$M_i = \begin{pmatrix} \sigma_l^i & 0 & 0 \\ 0 & \sigma_t^i & 0 \\ 0 & 0 & \sigma_n^i \end{pmatrix} \quad \text{and} \quad M_e = \begin{pmatrix} \sigma_l^e & 0 & 0 \\ 0 & \sigma_t^e & 0 \\ 0 & 0 & \sigma_n^e \end{pmatrix},$$

where σ_l, σ_t and σ_n are the conductivity values for each direction in the intracellular and extracellular domains.

As mentioned, the purpose of this thesis is to compare solutions to identify various break up patterns of wave propagation. The most important attribute when deciding on which mathematical model to use, is its ability to accurately portray various fibrillatory patterns. With this in mind, it is important that the model produces results that can be easily visualized. Hence, we let Ω be the unit square in two dimensions, i.e.:

$$\Omega = \{(x, y) \in \mathbb{R}^2 \, : \, 0 \leq x, y \leq 1\}.$$

This simplification of the domain Ω also eases the implementation of the solver. To generate a natural speed of the propagating wave, we adjust the capacitance of the tissue, C_m. For more details, see Section 4.1.

The level of physiological detail offered by differentiating intra- and extracellular conductivity is unnecessary and computationally exerting. By assuming equal anisotropic rates for the intra- and extracellular domain, i.e. $M_e = \lambda M_i$, for some scalar λ, we can simplify the bidomain equations (2.4) into a single partial differential equation.

Assuming $M_e = \lambda M_i$, then

$$\begin{aligned} \nabla(M_i \nabla v_e) &= -\frac{1}{1+\lambda} \nabla \cdot (M_i \nabla v), \\ \lambda \nabla \cdot (M_i \nabla v_e) &= -\chi I_m - I_{se}, \end{aligned}$$

which results in the following simple equation:

$$\frac{\lambda}{1+\lambda}\nabla \cdot (M_i \nabla v) = \chi I_m.$$

This is known as the monodomain equation and defines the monodomain model. The boundary terms with equal anisotropic rates are

$$\mathbf{n} \cdot (M_e \nabla v_e) = \mathbf{n} \cdot (\lambda M_i \nabla v_e) = 0,$$

and since $M_i \neq 0$ and $\lambda \neq 0$, we have the Neumann boundary conditions:

$$\frac{\partial v}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega. \tag{2.6}$$

As extracellular potential v_e has been removed from the equations, the boundary conditions of (2.5) is superfluous, resulting in a single intact boundary.

The assumption of equal anisotropic rates contradicts the physiological measurements of extracellular and intracellular conductivity. This makes it difficult to specify a parameter λ that obtains a good approximation to the underlying biophysiological behavior. Additionally, some important electrophysiological phenomena vanish with the assumption of equal anisotropy rates [5]. However, the monodomain model is not without its merits. It is considerably more compliant than the bidomain model when it comes to mathematical analysis and computation. Since the biophysiological accuracy is of lesser importance, the computational and analytical advantages provided by the monodomain model outweighs the accuracy given by the bidomain model.

2.2 Aliev Panfilov Two Variable Cell Model

In the equation (2.3) the transmembrane current density is given by

$$I_m = C_m \frac{\partial v}{\partial t} + I_{ion},$$

where I_{ion} is the ionic current given by some ionic model. It is common practice to examine these models in terms of single cell simulation. In the

sense of modeling single cells, the charge transported by the ionic current accumulated at the membrane affects the transmembrane potential as follows:

$$C_m \frac{\partial v}{\partial t} = -I_{ion} + I_s.$$

Here I_s denotes externally applied stimulus, which triggers the action potential in the cell. The models portraying the ionic current vary in biological accuracy and are chosen in accordance to the physiological behavior of interest. These ionic models can generally be grouped into three different categories [5]:

1. Phenomenological models, which are constructed to reproduce the macroscopically observed cell behavior. These are the simplest of the ionic models.

2. First generation models. These attempt to describe both the observed cellular behavior and the underlying physiology. These models reproduce the ionic currents that are most important for the action potential, and uses a simplified formulation of the underlying physiological process.

3. Second generation models offer a very detailed description of cell physiology. The models are based on advanced experimental techniques, enabling fine-scaled observations of the cell physiology.

As mentioned in Section 2.1, our main concern is accurate representation of the action potential and the different break up patterns. The physiological properties of the tissue in terms of cellular behavior should be accurate in the sense of action potential. The FitzHugh-Nagumo models is a set of first generation models which permit analytical estimation, and are usually numerically efficient for studying two- and three dimensional pulse dynamics in cardiac tissue. The models are successful in describing the qualitative aspects of the excitation propagation. However, they fall short when simulating several quantitative parameters of cardiac tissue, especially when modeling the shape of the action potential and the restitutional properties of the tissue.

The AlievPanfilov model is known for giving an accurate representation of the action potential and fibrillatory patterns, much thanks to the restitu-

tional property of the model [2]. The model it self consists of two equations characterizing the fast and slow process of depolarization and repolarization:

$$\begin{aligned} \frac{\partial v}{\partial t} &= -(v_{peak} - v_{rest})(kV(V-a)(V-1) + Vs) + I_s, \\ \frac{\partial s}{\partial t} &= 0.25\epsilon(v,s)(-s - kV(V-a1)). \end{aligned} \tag{2.7}$$

Here $\epsilon(v,w) = \epsilon_0 + \mu_1 w/(v+\mu_2)$ and $V = (v - v_{rest})/(v_{peak} - v_{rest})$. The parameters k, a, ϵ_0 are given and may be adjusted to simulate different cell types. v_{peak} is the highest value of the transmembrane potential, while v_{rest} is the membrane potential for cell at resting state. The variable s is the recovery potential initiating the repolarization process of the action potential. The parameters μ_1 and μ_2 are parameters governing the restitutional properties of the tissue and will be regulated to simulate different repolarization phases. Changes to μ_1 and μ_2 directly affects the action potential duration and cycle length [2]. In 4, we will see how different fibrillatory patterns can be constructed by varying the parameter μ_1.

2.3 Mathematical Norms

Let X be a vector space over some field $\mathbb{F}$. A *norm* on X, as defined in [6], is a function $\|\cdot\| : X \longrightarrow \mathbb{R}$ such that for all $x, y \in X$,$\alpha \in \mathbb{F}$,

i) $\|x\| \geq 0$

ii) $\|x\| = 0 \Leftrightarrow x = 0$

iii) $\|\alpha x\| = |\alpha|\|x\|$

iv) $\|x + y\| \leq \|x\| + \|y\|$

If $X = \mathbb{R}^n$, then,

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2 + \cdots + |x_n - y_n|^2},$$

is the Euclidean norm. This norm gives the shortest distance between two two points x, y in Euclidean geometry. Similarly, it might be possible to construct a norm to measure the distance between fibrillatory and stable solutions of cardiac simulations in some function space. We will therefore apply a few well-known norms to different fibrillatory simulations, to see if

they have some particular qualities to numerically differentiate these from laminar flows.

A vector space X on which there is a norm, is called a normed vector space, or just a normed space [6]. In this thesis we will look at some of the most well-known normed spaces, namely the Lebesgue and Sobolev spaces. More specifically, we will be looking at the Lebesgue space $L^2(\Omega)$ and the Sobolev spaces $H^1(\Omega)$ and $H^2(\Omega)$. The norms defining these spaces are

$$\|f\|_{L^2(\Omega)} = \left(\int_\Omega |f\mathbf{x}|^2 \, d\mathbf{x}\right)^{\frac{1}{2}}, \tag{2.8}$$

$$\|f\|_{H^1(\Omega)} = \left(\|f\|_{L^2(\Omega)^2} + \|\nabla f\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}, \tag{2.9}$$

$$\|f\|_{H^1(\Omega)} = \left(\|f\|_{L^2(\Omega)^2} + \|\nabla f\|^2_{L^2(\Omega)} + \|\nabla^2 f\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}, \tag{2.10}$$

These are the most common variants among the Lebesgue and Sobolev space. One reason for their popularity, is that they have a very much sought after property, they are *Banach*. A space which is Banach is a complete space in which every Cauchy sequnece converges.

As it might not be obvious that these norms actually satisfies all the axioms of a norm, we will show that they do. Note however, that it is enough to only show that L^2-norm satisfies all the axioms, as the H^1- and H^2-norms are variants of the L^2-norm, and the same argmuents apply to them. Supposing Ω is non-empty, then

$$0 = \|f\|^2_{L^2(\Omega)} = \int_\Omega |f(\mathbf{x})|^2 \, d\mathbf{x} \Rightarrow f = 0 \, \forall \mathbf{x} \in \Omega,$$

and if $f(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Omega$ then

$$\|f\|^2_{L^2(\Omega)} = \int_\Omega |f(\mathbf{x})|^2 \, d\mathbf{x} = 0.$$

The axiom ii) is trivial as $|f| \geq 0$. For iii), let $\alpha \in \mathbb{F}$. Then

$$\|\alpha f\|_{L^2(\Omega)} = \left(\int_\Omega |\alpha f(\mathbf{x})|^2 \, d\mathbf{x}\right)^{\frac{1}{2}} = |\alpha| \left(\int_\Omega |(\mathbf{x}|^2 \, d\mathbf{x}\right)^{\frac{1}{2}}.$$

Lastly, suppose $f, g : \Omega \longrightarrow \mathbb{F}^n$. Then we get

$$\begin{aligned}
\|f+g\|^2_{L^2(\Omega)} &= \int_\Omega |f(\mathbf{x})+g(\mathbf{x})|^2\, d\mathbf{x} \\
&\leq \int_\Omega |f(\mathbf{x})+g(\mathbf{x})|(|f(\mathbf{x})|+|g(\mathbf{x})|)\, d\mathbf{x} \\
&\leq \|f+g\|_{L^2(\Omega)} \left(\left(\int_\Omega |f(\mathbf{x})|^2\, d\mathbf{x} \right)^{\frac{1}{2}} + \left(\int_\Omega |g(\mathbf{x})|^2\, d\mathbf{x} \right)^{\frac{1}{2}} \right) \\
&= \|f+g\|_{L^2(\Omega)} (\|f\|_{L^2(\Omega)} + \|g\|_{L^2(\Omega)}).
\end{aligned}$$

Thereby it is shown that the L^2-norm, given by (2.8), satisfies all norm criteria.

Another type of measurement used on functions are semi-norms [6]. For some vector space X, the semi-norm on X is a real-valued function $p : X \longrightarrow \mathbb{R}$, such that

i) $p(x+y) \leq p(x) + p(y) \quad x, y \in X$

ii) $p(\alpha x) = |\alpha| p(x) \quad x \in X, \alpha \in \mathbb{F}$

The semi-norm is weaker in the sense that $p(x) = 0$ does not necessarily imply that $x = 0$. Total variation is a semi-norm, and is defined as

$$V(f, \Omega) = \int_\Omega |\nabla f(\mathbf{x})|\, d\mathbf{x},$$

It is clear that

$$V(f+g, \Omega) = \int_\Omega |\nabla(f+g)(\mathbf{x})|\, d\mathbf{x} \leq \int_\Omega |\nabla f(\mathbf{x})|\, d\mathbf{x} + \int_\Omega |\nabla g(\mathbf{x})|\, d\mathbf{x},$$

and

$$V(\alpha f, \Omega) = \int_\Omega |\alpha \nabla f(\mathbf{x})|\, d\mathbf{x} = |\alpha| \int_\Omega |\nabla f(\mathbf{x}|\, d\mathbf{x}.$$

However, unlike for norms, $V(f, \Omega) = 0$ simply implies that $\nabla f = 0$. For a real-valued function f on an interval $[a, b] \subset \mathbb{R}$, the total variation defines the measure of the one-dimensional arc length of the curve. Similarly, we hope find some features on the total variation of the solutions obtained in Section 4.2, which can distinguish between different fibrillatory patterns.

Chapter 3

Numerical Methods

3.1 The Explicit Model

Originally, we wanted to use an explicit solver for the monodomain model for simulating the electrical behavior in cardiac tissue. Explicit methods for solving partial- and ordinary differential equations are simple and straightforward to implement. Since the physiological accuracy is of minor importance, we concluded that a first order method would suffice. However, the problem with a explicit solver is the strict stability conditions put on Δt, especially by the finite difference method. These requirements forces Δt to be very small, generating a vast number iteration, making the simulator numerically inefficient. This will be discussed throughout the following section, particularly in Section 3.1.3.

3.1.1 Godunov Splitting

Solving nonlinear partial differential equations like the monodomain equation can be a difficult task. Operator splittings are amongst techniques that simplifies these nonlinear problems. The method given below is the Godunov splitting [5]. Utilizing Godunov splitting, we formulate the monodomain equation, (2.1), in terms of operators:

$$\mathcal{L}_1 v = -\frac{1}{C_m} I_{ion}(v, w) \tag{3.1}$$

$$\mathcal{L}_2 v = \frac{\lambda}{\chi C_m (1+\lambda)} \nabla \cdot (M_i \nabla v) \tag{3.2}$$

$$\mathcal{K} s = 0.25\epsilon(v, s)(-s - kV(V - a - 1)) \tag{3.3}$$

With these operators, we define the following equations:

$$\begin{aligned} \mathcal{L}_1 w &= \frac{\partial w}{\partial t}, \\ \mathcal{K} s &= \frac{\partial s}{\partial t}, \\ w(t_n) &= v(t_n) \end{aligned} \tag{3.4}$$

and

$$\begin{aligned} \mathcal{L}_2 u &= \frac{\partial v}{\partial t}, \\ u(t_n) &= w(t_n + \Delta t). \end{aligned} \tag{3.5}$$

Each equation is solved on the interval $t \in [t_n, t_n + \Delta t]$. The solution to Equation (3.4), $w(t_n + \Delta t)$, is used as the initial value for Equation (3.5), while the solution $u_(t_n + \Delta t)$ gives the solution to Equation (2.1) for a single time step, i.e. $v(t_n + \Delta t)$.

Utilizing the Godunov splitting reduces the difficult nonlinear partial differential equation into a system of coupled equations: a linear partial differential equation and an ordinary differential equation. It may seem like the solution has been estimated for an interval of $2\Delta t$. However, only some parts of the monodomain equation (2.1) are included in each calculation.

Performing a Taylor series expansion on the solution of Equation (2.1), $v(t_{n+1})$, and comparing it to the approximate solution $u(t_n + \Delta t)$, we will see that the result is in fact a consistent approximation. The Taylor expansion of the exact solution of Equation (2.1) at time $t = t_n + \Delta t$, is given by

$$v(t_{n+1}) = v(t_n) + \Delta t \frac{\partial v}{\partial t}|_{t=t_n} + \Delta t^2 \frac{\partial^2 v}{\partial t^2}|_{t=t_n} + \mathcal{O}(\Delta t). \tag{3.6}$$

In terms of operators (3.2) and (3.1), the monodomain equation can be written as follows:

$$\frac{\partial v}{\partial t} = (\mathcal{L}_1 + \mathcal{L}_2)v.$$

Furthermore, since neither $\mathcal{L}_1$ or $\mathcal{L}_2$ is explicitly dependent on t, then by direct differentiation

$$\frac{\partial^k v}{\partial t^k} = (\mathcal{L}_1 + \mathcal{L}_2)^k v.$$

The notation $(\mathcal{L}_1 + \mathcal{L}_2)^k$ implies that the operator $(\mathcal{L}_1 + \mathcal{L}_2)$ is applied k times. For more details, see [5]. Hence, writing (3.6) in terms of operators, we get:

$$v(t_{n+1}) = v(t_n) + \Delta t(\mathcal{L}_1 + \mathcal{L}_2)v(t_n) + \Delta t^2(\mathcal{L}_1 + \mathcal{L}_2)^2 v(t_n) + \mathcal{O}(\Delta t^3). \quad (3.7)$$

Similarly, by Taylor expanding the solution of Equation (3.4), we get

$$w(t_n + \Delta t) = v(t_n) + \Delta t\mathcal{L}_1 v(t_n) + \Delta t^2 \mathcal{L}_1^2 v(t_n) + \mathcal{O}(\Delta t^3). \quad (3.8)$$

The Taylor series for the solution of Equation (3.4), u, can be written as

$$u(t_n+\Delta t) = wt_n+\Delta t)+\Delta t\mathcal{L}_2 w(t_n+\Delta t)+\Delta t^2\mathcal{L}_2^2 u(t_n+\Delta t)+\mathcal{O}(\Delta t^3). \quad (3.9)$$

If we include the expression found in (3.8) for the initial value $w(t_n+\Delta t)$ in Equation (3.9), this gives us

$$u(t_n+\Delta t) = v(t_n)+\Delta t(\mathcal{L}_1+\mathcal{L}_2)v(t_n)+\Delta t^2(\mathcal{L}_1^2+2\mathcal{L}_2\mathcal{L}_1+\mathcal{L}_1^2)v(t_n)+\mathcal{O}(\Delta t^2). \quad (3.10)$$

Then, by examining the difference between (3.7) and (3.10), we see that

$$\begin{aligned} w(t_n + \Delta t) - v(t_{n+1}) &= \frac{\Delta t^2}{2}(\mathcal{L}_1\mathcal{L}_2 + \mathcal{L}_2\mathcal{L}_1)v(t_n) + \mathcal{O}(\Delta t^3) \\ &= \mathcal{O}(\Delta t^2). \end{aligned}$$

This shows that the Godunov splitting gives a consistent approximation to v with an *a priori* error estimate of $\mathcal{O}(\Delta t^2)$ for the interval $[t_n, t_{n+1}]$.

It is shown that the error of each discrete time interval $[t_n, t_{n+1}]$ is proportional to Δt^2. Moreover, we see that the error accumulates to $n\Delta t^2$ after n time steps. Then, solving the monodomain equation for a fixed time interval, $t \in [0, T]$, the number of intervals N is proportional to Δt^{-1}. This gives an *a priori* error estimate of $\mathcal{O}(\Delta t)$ at $t = T$, and further gives the notion that the Godunov splitting has a first-order time accuracy.

3.1.2 Euler Method

When solving Equation (3.4) on some interval $t \in [t_n, t_{n+1}]$, we integrate the equation on both sides, such that

$$\int_{t_n}^{t_{n+1}} \frac{\partial w}{\partial t} = \int_{t_n}^{t_{n+1}} \mathcal{L}_2 w(t)\, dt.$$

$w(t_n + \Delta t)$ is satisfied by the equation

$$w(t_n + \Delta t) = w(t_n) + \int_{t_n}^{t_{n+1}} \mathcal{L}_2 w(t)\, dt. \tag{3.11}$$

In most cases, the integral on the right hand side is quite hard to compute analytically and must be approximated. The approximation of the integral in equation (3.11) is in many ways what results in the precision of the numerical method being used.

The forward Euler method is a simple numerical method for solving ordinary differential Equations [5]. The forward Euler method estimates the integral by assuming $w(t) = w(t_n)$, and then setting $\mathcal{L}_2 w(t) = C$, where $C \in \mathbb{R}$, and thereby we obtain the following approximation:

$$w(t_n + \Delta t) = w(t_n) + C \int_{t_n}^{t_{n+1}} dt = w(t_n) + \Delta t \mathcal{L}_2 w(t).$$

This is a very rough approximation, but is computationally efficient.

By Taylor expanding the actual solution to Equation (3.4), it is easy to see that the *a priori* error estimate accumulates to $n\Delta t^2$. Using the same argument as for the Godunov splitting when solving Equation (3.4) on an interval $t \in [0, T]$, the number of intervals N are proportional to Δt^{-1}, leading to the well-known fact that the forward Euler is a first order accuracy method.

One of the drawbacks using the forward Euler method, is the poor stability. The stability function is defined as $R(z) := 1 + z$, where $z = \lambda \Delta t$ and the value λ is the Eigenvalues of the Jacobian matrix [5]. This gives the stability domain

$$S = \{z \in \mathbb{C} : |1 + z| \leq 1\},$$

which results in the forward Euler being stable as long as $-2 \leq \lambda \Delta t \leq 0$. It is clear that the the stability of the solver is very dependent on the Eigenvalues.

Eigenvalues

In figures 3.1-3.5 we see the Eigvenvalues to Jacobian matrix functions of s and v. The obtained Eigenvalues are dependent on the parameter μ_1 (see Section 4). Thus in Section 4.2, we had to construct one plot for each μ_1. s is plotted on the x-axis, v is plotted on the y-axis. In terms of stability, we are interested in finding the smallest negative Eigenvalues, as areas with positive Eigenvalues are areas results from unstable break up of the action potential. These positive Eigenvalues are generated by the rapid depolarization process.

The red line marked by the $\circ$, defines the domain where the solutions s and v are located. The smallest Eigenvalue within the solution domain is for all simulations $\lambda = -2.5$. This results in the following stability restriction on Δt:

$$\Delta t \leq 0.85,$$

which is a sufficiently large time step. However, as we will see in Section 3.1.3, the estimate on Δt will become considerably more strict.

3.1.3 Explicit Finite Difference Method

The partial differential equation of the Godunov splitting, can been solved with an explicit finite difference method [1]. The finite difference method utilizes approximations of derivatives, by combining nearby function values, using a set of weights. In one dimension, the finite difference approximation to the second derivative of u, can be found by considering the Taylor series

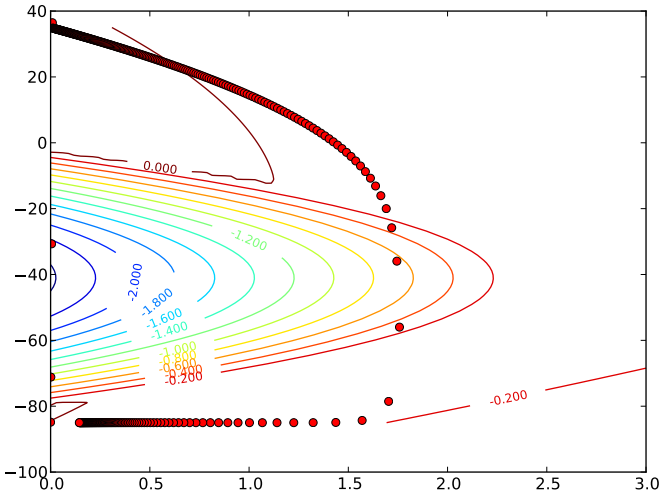

Figure 3.1: The smallest Eigenvalues of the Jacobian matrix for simulation with $\mu_1 = 0.07$

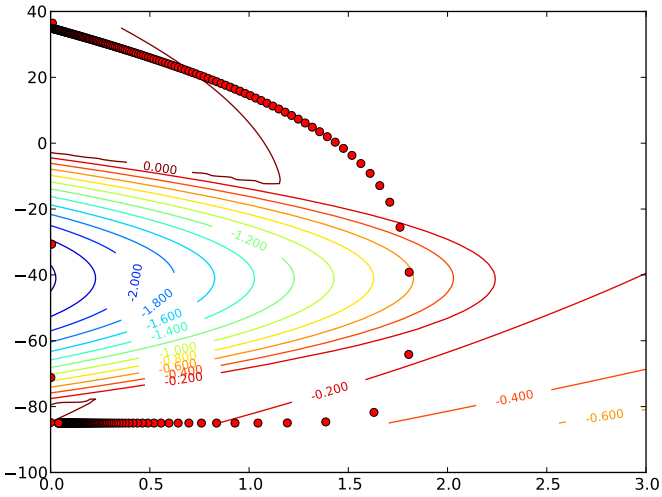

Figure 3.2: The smallest Eigenvalues of the Jacobian matrix for simulation with $\mu_1 = 0.14$

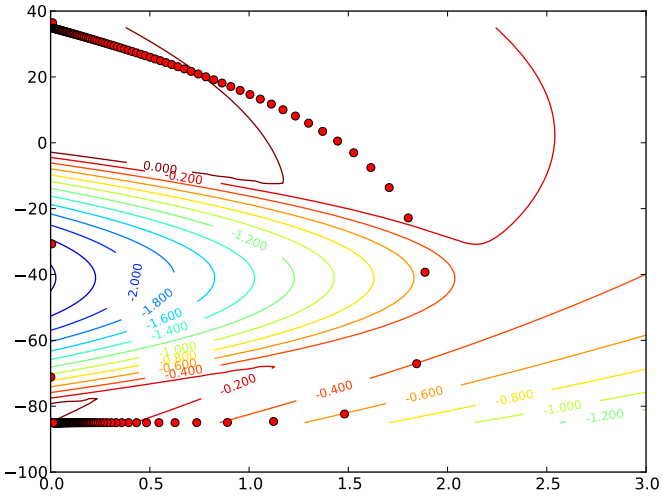

Figure 3.3: The smallest Eigenvalues of the Jacobian matrix for simulation with $\mu_1 = 0.28$

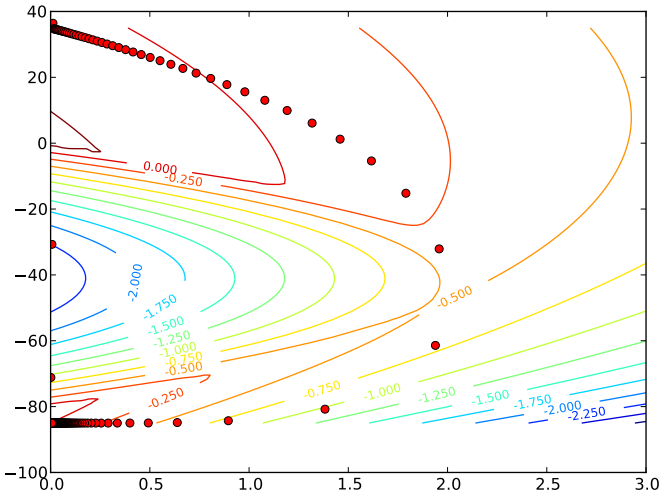

Figure 3.4: The smallest Eigenvalues of the Jacobian matrix for simulation with $\mu_1 = 0.56$

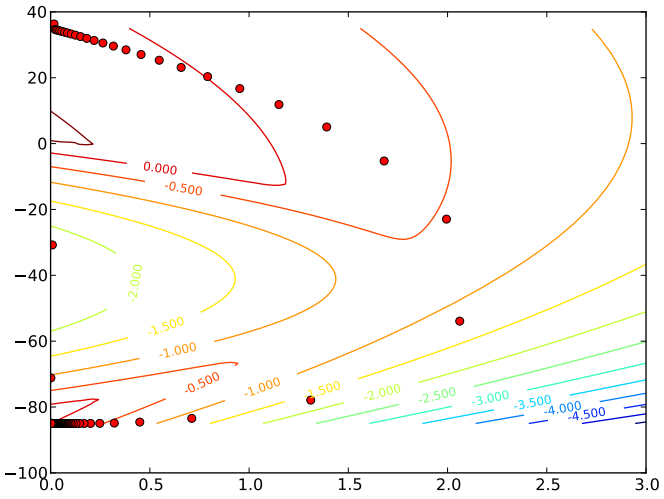

Figure 3.5: The smallest Eigenvalues of the Jacobian matrix for simulation with $\mu_1 = 1.12$

$$u(x + \Delta x) = u(x) + \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial u}{\partial x} + \mathcal{O}(\Delta x^3),$$

and

$$u(x - \Delta x) = u(x) - \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial u}{\partial x} + \mathcal{O}(\Delta x^3).$$

Adding the two equation, we find that

$$\frac{\partial^2 u}{\partial x^2} = \frac{-2u(x) + u(x - \Delta x) + u(x + \Delta x)}{\Delta x^2} + \mathcal{O}(\Delta x^2), \tag{3.12}$$

yielding a second order spatial estimate of the second derivative. With a similar argument as for the forward Euler we have that

$$u(t_n + \Delta t) = u(t_n) + \Delta t \mathcal{L}_2 u(t_n) + \mathcal{O}(\Delta t^2). \tag{3.13}$$

From these estimates we can see that the the finite difference method will have second-order accuracy in space, and only first-order accuracy for time discretization. Note that by reducing $\mathcal{O}(\Delta t^2)$ to $\mathcal{O}(\Delta t)$, the estimation of $\partial u / \partial t$ follows the same line of argument as made for Godunov splitting and forward Euler.

The construction of the finite difference scheme for Equation (3.5) depends on the specified domain. As mentioned in Section 2.1, we will consider the domain as a the unit square:

$$\Omega := \{(x, y) \in \mathbb{R}^2 \, : \, 0 \leq x, y \leq 1\}.$$

Fibrillatory action potential is easily recognizable and very characteristic on a two dimensional plane. Also, the rectangular shape of the domain synergises well with the finite difference method. The actual shape of the cardiac muscle should have little consequence for the norm comparisons. The fact that the square is unitary is of no regard, as propagation of the action potential can be regulated by modifying the electrical capacitance, C_m, of the tissue. The unit square is easily discretizised by

$$\Omega_h := \{(x_i, y_j) \, : \, 0 \leq i, j \leq \text{ for } i, j \in \mathcal{N}\}. \tag{3.14}$$

Here $(x_i, y_j) := (i\Delta x, j\Delta y)$, where $h = 1/m$ defines the distance between spatial grid points in an $m \times m$ mesh of Ω. For simplicity, we let $\Delta x = \Delta y = h$.

For the two dimensional case, the Equation (3.5) is given by

$$\frac{\partial u}{\partial t} = \frac{\lambda}{\chi C_m(1+\lambda)} \left[\sigma_l^i \frac{\partial^2 u}{\partial x^2} + \sigma_t^i \frac{\partial^2 u}{\partial y^2} \right]. \tag{3.15}$$

We denote $u(x_i, y_j, t_n) := u_n^{i,j}$. Using the derivative estimates obtained above ((3.13) and (3.12)) we construct the so-called five-point stencil for the finite difference scheme. Then the discrete solution to equation 3.15 on the interior of the domain, $\Omega_h / \partial\Omega_h$, is given by

$$u_{n+1}^{i,j} = (1 - 2r\sigma_l^i - 2r\sigma_t^i)u_n^{i,j} + r\sigma_l^i(u_n^{i+1,j} + u_n^{i-1,j}) + r\sigma_t^i(u_n^{i,j+1} + u_n^{i,j-1}). \tag{3.16}$$

For convenience of notation, we introduce

$$r = \frac{\lambda \Delta t}{\chi C_m (1+\lambda) h^2}. \tag{3.17}$$

The solution on the boundary of Ω differs slightly from Equation (3.16), due to the Neumann boundary conditions. Since $\partial u / \partial \mathbf{n} = 0$, we incorporate

some auxiliary grid points to Ω_h. The auxiliary nodes are defined by the equations

$$\frac{u_n^{-1,j} - u_n^{1,j}}{2h} = 0 \quad \frac{u_n^{m,j} - u_n^{m+1,j}}{2h} = 0, \tag{3.18}$$

and

$$\frac{u_n^{i,-1} - u^{i,1}}{2h} = 0 \quad \frac{u_n^{i,m} - u_n^{i,m+1}}{2h} = 0. \tag{3.19}$$

Then, letting $u_n^{i,-1} = u_n^{i,1}, u_n^{i,m+1} = u_n^m, u_n^{-1,j} = u_n^{1,j}$ and $u_n^{m,j} = u_n^{m+1,j}$, we use Equation (3.16), giving the solutions u_{n+1} on $\partial\Omega_h$.

The explicit finite difference method was deemed non-viable. The strict stability requirements imposed by it [1],

$$\frac{\Delta t}{\Delta h^2} \leq \frac{1}{2},$$

made the solver numerically inefficient, forcing an extremely small Δt. With a 200×200 mesh, then $h = 1/200$, resulted in 80000 iterations for each millisecond of simulation. Considering that we are running simulations on the interval $t \in [0ms, 1000ms]$, we can easily conclude that the vast number of iterations makes the solver too slow for practical use.

3.2 The Semi-Implicit Model

As described in Section 3.1, the stability condition offered by the mentioned solvers are much too strict. In this section we will examine some models with more relaxed stability requirements. However, since the requirements are less strict on Δt, we will employ methods with second-order time accuracy to warrant an even larger Δt. This way we are able to fully exploit the benefits of loose stability conditions.

We will still be utilizing operator splitting to deal with the non-linearity of the monodomain equation, but instead of the first order Godunov splitting, we apply the slightly different Strang splitting. For the partial derivative part of the Strang splitting [5], we employ a semi-implicit finite difference method with a Crank-Nicolson scheme. The ordinary differential equation

is still solved explicitly, but to achieve second-order time accuracy, we utilize a Runge-Kutta method [11].

3.2.1 Strang Splitting

The Strang splitting algorithm is very similar to the Godunov splitting [5]. The main difference is that the Strang splitting offers second-order accuracy by including an intermediate time step. As described earlier, this splitting technique divides the monodomain equation into the operators

$$\mathcal{L}_1 = -\frac{1}{C_m} I_{ion}, \tag{3.20}$$

$$\mathcal{L}_2 = \frac{\lambda}{\chi C_m (1+\lambda)} \nabla \cdot (M_i \nabla v), \tag{3.21}$$

$$\mathcal{K} = 0.25\epsilon(v,s)(-s - kV(V-a-1)), \tag{3.22}$$

using each operator to solve a coupled set of equations. However, while the Godunov splitting solved the Equations (3.4) and (3.5) on the full length of the interval $[t_n, t_n + \Delta t]$, the Strang splitting incorporates an additional step to the algorithm. The Strang splitting can be described as a three step algorithm:

(i) First we solve

$$\begin{aligned} \frac{\partial w}{\partial t} &= \mathcal{L}_1 w, \\ \frac{\partial s}{\partial t} &= \mathcal{K} s, \\ w(t_n) &= v(t_n) \end{aligned} \tag{3.23}$$

for $t \in [t_n, t_n + \Delta t/2]$.

(ii) Then,

$$\begin{aligned} \frac{\partial u}{\partial t} &= \mathcal{L}_2 u, \\ u(t_n) &= w(t_n + \Delta t/2), \end{aligned} \tag{3.24}$$

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

(iii) Finally we solve the problem

$$\begin{aligned} \frac{\partial w}{\partial t} &= \mathcal{L}_1 w, \\ \frac{\partial s}{\partial t} &= \mathcal{K} s, \\ w(t_n + \Delta t/2) &= u(t_n + \Delta t), \end{aligned} \tag{3.25}$$

for the remainder of the interval $t \in [t_n + \Delta t/2, t_n + \Delta t]$. Further, the solution to the final equation is set to $w(t_{n+1}) = v(t_{n+1})$

To show that the Strang splitting truly gives $\mathcal{O}(\Delta t^2)$ precision, consider the Taylor series expansion for w in Equation (3.23) with the initial value $v(t_n)$;

$$w(t_n + \Delta/2) = v(t_n) + \frac{\Delta t}{2}\mathcal{L}_1 v(t_n) + \frac{\Delta t^2}{4}\mathcal{L}_1^2 v(t_n) + \mathcal{O}(\Delta t^3)$$

This is the initial value for the Equation (3.24). Thus the solution $u(t_n + \Delta t)$ is given by:

$$\begin{aligned} u(t_n + \Delta t) = v(t_n) + \Delta t \left(\frac{\mathcal{L}_1}{2} + \mathcal{L}_2\right) v(t_n) + \\ \frac{\Delta t^2}{2}\left(\frac{\mathcal{L}_1^2}{4} + \mathcal{L}_2\mathcal{L}_1 + \mathcal{L}_2^2\right) v(t_n) + \mathcal{O}(\Delta t^3). \end{aligned}$$

Lastly, by Taylor expanding the solution of Equation (3.25) with the initial value found above, we get

$$\begin{aligned} w(t_n + \Delta t) = v(t_n) + \Delta t \left(\mathcal{L}_1 + \mathcal{L}_2\right) v(t_n) + \\ \frac{\Delta t^2}{2}\left(\mathcal{L}_1^2 + \mathcal{L}_1\mathcal{L}_2 + \mathcal{L}_2\mathcal{L}_1 + \mathcal{L}_2\right) v(t_n) + \mathcal{O}(\Delta t^3). \end{aligned}$$

Using the Taylor series from Equation (3.7), we estimate

$$v(t_n + \Delta t) - w(t_n + \Delta t) = \mathcal{O}(\Delta t^3.)$$

With the same argument as for Godunov splitting, we have that the local error is proportional to $\mathcal{O}(\Delta t^3)$, and the accumulate error after N intervals is equal to Δt^{-1}. Hence, the *a priori* error estimate on the Strang splitting

is $\mathcal{O}(\Delta t^2)$.

3.2.2 Runge Kutta Methods

For an ordinary differential equation on the form

$$\frac{\partial w}{\partial t} = f(w), \tag{3.26}$$

the Runge-Kutta methods provide a numerical approximation to the solution of these equations [11]. The general discrete solutions of equation (3.26) is given by the Runge-Kutta methods as:

$$y_i = w_n + \Delta t \sum_{j=1}^{s} k_{i,j} f(y_j) \quad \text{forÂ } 1 \leq i \leq s+1, \tag{3.27}$$

$$w_{n+1} = y_{s+1}. \tag{3.28}$$

The parameters $k_{i,j} \in \mathbb{R}$ and specifies the method being used. The variables y_i are intermediate estimates used for computing w_{n+1}. The solution scheme becomes implicit if at least one coefficient $k_{i,j} \neq 0$ for $j \leq i$. However, due to the Eigenvalues obtained in section 3.1.2, we conclude that there will be little to gain computationally by implementing an implicit solver, as these tend to be numerically strenuous.

By letting all $k_{i,j} = 0$ for $j \leq i$, we generate an explicit method. For our solver we want to use a second-order method. By choosing $k_{2,1} = 1$ and $k_{3,1} = k_{3,2} = 1/2$ for $s = 3$, we get a $\mathcal{O}(\Delta t^3)$ approximation to the solution on a single interval.

Consider the Taylor series

$$f(w_n + \Delta t f(w_n)) = f(w_n) + \Delta t \frac{\partial f}{\partial w} f(w_n) + \mathcal{O}(\Delta t^2). \tag{3.29}$$

Note that

$$\frac{\partial^2 w}{\partial t^2} = \frac{\partial f}{\partial t} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial t} = \frac{\partial f}{\partial u} f.$$

Hence, the Taylor series from equation (3.29) is given as

$$f(w_n + \Delta t f(w_n)) = f(w_n) + \Delta t \frac{\partial^2 w}{\partial t^2} + \mathcal{O}(\Delta t^2),$$
$$= \frac{\partial w}{\partial t} + \Delta t \frac{\partial^2 w}{\partial t^2} + \mathcal{O}(\Delta t^2).$$

The discrete solution to equation (3.26) given by the coefficient $k_{2,1} = 1$ and $k_{3,1} = k_{3,2} = 1/2$, is further defined as

$$w_{n+1} = w_n + \frac{\Delta t}{2} y_1 + \frac{\Delta t}{2} y_2$$
$$= w_n + \frac{\Delta t}{2} \frac{\partial w}{\partial t} + \frac{\Delta t}{2} \frac{\partial w}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 w}{\partial t} + \mathcal{O}(\Delta t^3),$$

which compared to the solution of equation (3.26) gives an approximation of $\mathcal{O}(\Delta t^3)$. As earlier, the error accumulates to $\mathcal{O}(n\Delta t)$, which after N intervals proportional to Δt^{-1} results in a second-order time discretization accuracy. This concludes that the Runge-Kutta method described by the coefficients $k_{2,1} = 1$ and $k_{3,1} = k_{3,2} = 1/2$ is a second-order solver.

The stability function is given as [5]

$$R(z) = 1 + z + \frac{z^2}{2},$$

where $z := \lambda \Delta t$, and λ are the eigenvalues of the Jacobian matrix of f. Similar to the forward Euler described above, this Runge-Kutta method is stable as long as $|R(z)| \leq 1$. Hence, Δt must satisfy $-2 \leq \lambda \Delta t(1 + \lambda \Delta t/2) \leq 0$. With the smallest Eigenvalue obtained in section 3.1.2, this results in the following estimate on Δt.

$$-2.4^2 \Delta t^2 + 2.4 \Delta t - 2 \leq 0$$

3.2.3 Semi-Implicit Finite Difference Method

The partial differential Equation (3.24) will be solved using a semi-implicit finite difference method with a Crank-Nicolson time step approximation. This is a popular technique for solving partial differential equations occurring in electrochemical kinetic modeling [3]. One of the major advantages

of the method is that it is unconditionally stable [9], thus Δt and Δx can be chosen independently.

The time scheme for the Equation (3.24) with a Crank-Nicolson estimate is given by

$$u(t_n + \Delta t) - \frac{\Delta t}{2}\mathcal{L}_2 u(t_n + \Delta t) = u(t_n) + \frac{\Delta t}{2}\mathcal{L}_2 u(t_n)). \tag{3.30}$$

The time accuracy can be shown to be second-order by considering the Taylor series

$$\mathcal{L}_2 u(t_n + \Delta t) = \mathcal{L}_2 u(t_n) + \Delta t \mathcal{L}_2 u(t_n) + \frac{\Delta t^2}{2}\mathcal{L}_2 u(t_n) + \mathcal{O}(\Delta t^3). \tag{3.31}$$

Including the Taylor series expression (3.31) for $\mathcal{L}_2 u(t_n + \Delta t)$ in Equation (3.30), we get

$$u(t_n + \Delta t) = u(t_n) + \Delta t \mathcal{L}_2 u(t_n) + \frac{\Delta t^2}{2}\mathcal{L}_2 u(t_n) + \mathcal{O}(\Delta t^3)$$

It is easy to see that the time error accumulates as $\mathcal{O}(\Delta t^3)$ for each time step, thus the collective error estimate of one complete simulation will be $\mathcal{O}(\Delta^2)$.

Applying the estimates on the second derivatives found in Equation (3.12), the one dimensional solution to the partial differential Equation (3.24) is given by

$$u^i_{n+1} + \Delta t \left(r u^i_{n+1} - \frac{r}{2} u^{i-1}_{n+1} - \frac{r}{2} u^{i+1}_{n+1} \right) = u^i_n + \Delta t \left(-r u^i_n + \frac{r}{2} u^{i-1}_n + \frac{r}{2} u^{i+1}_n \right), \tag{3.32}$$

where $u^i_n = u(n\Delta t, i\Delta x)$ and similar to constants (3.17)

$$r = \frac{\lambda}{\chi C_x (1 + \lambda) \Delta x^2}.$$

By regarding each discrete value $u^{i,j}$ as entries in a $m \times m$ matrix,

$$U = \begin{pmatrix} u^{1,1} & \cdots & u^{1,m} \\ \vdots & \ddots & \vdots \\ u^{m,1} & \cdots & u^{m,m} \end{pmatrix},$$

the problem (3.32) is solved by the linear equation:

$$(I - \Delta t A)U_{n+1} = (I + \Delta t A)U_n = b, \tag{3.33}$$

for the interior of the domain. From Equation (3.32), it is i easy to see that A must be tridiagonal matrix with $-r$ on the diagonal and $r/2$ on the off-diagonal.

In the one dimensional case, the grid points in the field U_{n+1} only receive a contribution from the horizontally neighboring grid points, while in two dimensions we also have to consider the vertically neighboring entries. This makes it slightly more difficult to construct the matrix A. The finite difference method with a Crank-Nicholson time scheme in two dimensions for the monodomain equation is given by

$$\begin{aligned} &u_{n+1}^{i,j} + \Delta t \sigma_l^i \frac{r}{2}(2u_{n+1}^{i,j} - u_{n+1}^{i-1,j} - u_{n+1}^{i+1,j})\Delta t \sigma_t^i \frac{r}{2}(2u_{n+1}^{i,j} - u_{n+1}^{i,j-1} - u_{n+1}^{i,j+1}) = \\ &u_n^{i,j} + \Delta t \sigma_l^i \frac{r}{2}(-2u_n^{i,j} + u_n^{i+1,j} + u_n^{i-1,j})\Delta t \sigma_t^i \frac{r}{2}(-2u_n^{i,j} + u_n^{i,j+1} + u_n^{i,j-1}). \end{aligned} \tag{3.34}$$

In principal it is very similar to the one dimensional case, in the sense that we solve a linear Equation on form (3.33). However, to account for both the horizontal and vertical grid points of the mesh U_n when finding U_{n+1}, we have to transform the u_n and u_{n+1} so that

$$\mathbf{u} = \begin{pmatrix} u^{m,1} \\ \vdots \\ u^{m,m} \\ \vdots \\ u^{1,m} \\ \vdots \\ u^{1,1} \end{pmatrix}.$$

Then, we solve the equation

$$(I - \Delta t A)\mathbf{u}_{n+1} = (I + \Delta t A)\mathbf{u}_n = b, \tag{3.35}$$

where A and I from Equation (3.33) are sparse $m^2 \times m^2$ matrices.

Looking at A in terms of $m \times m$ block matrices while neglecting the boundary conditions, we see from Equation (3.34) that for each i

$$\begin{aligned} &u^{i,i} - A^{i,i-1}u_{n+1}^{\cdot,i-1} - A^{i,i}u_{n+1}^{\cdot,i} - A^{i,i+1}u_{n+1}^{\cdot,i+1} = \\ &u^{i,i} + A^{i,i-1}u_{n+1}^{\cdot,i-1} + A^{i,i}u_{n+1}^{\cdot,i} + A^{,i+1}u_{n+1}^{\cdot,i+1}. \end{aligned}$$

Hence, all block matrices $A^{i,i-1} = A^{i,i+1}$ must be diagonal matrices with $\sigma_l^i r/2$ on the diagonal. From the one dimensional instance we can conclude that all the diagonal block matrices $A^{i,i}$ must be the tridiagonal matrix with $-\sigma_l^i r - \sigma_t^i r$ on the diagonal and $\sigma_t^i r/2$ on the off-diagonal. However, because of the Neumann boundary conditions, the entries $(1,2)$ and $(m, m-1)$ of each $A^{i,i}$ must be counted twice, i.e. these entries will be $\sigma_t^i r$, accounting for the end points of each column. Similarly, we get $A^{1,2} = A^{m,m-1} = 2A^{i,i-1} = 2A^{i,i+1}$ for $i = 2, \ldots, m-1$. For details on how to solve the linear Equation (3.35), see appendix A.1.

3.2.4 *A Posteriori* Error Estimate

We have examined the *a priori* for each method used. In this section, we will check if this estimate correlates to the *a posteriori* error estimate. Solving the monodomain equation on the interval $t \in [0\text{ms}, 6\text{ms}]$, we ran several simulations using different values for Δt. For each solution we applied an external stimulus $i_s = 15\text{mV/ms}$ lasting from 0ms to 3ms. The finest time discretization used was $\Delta t = 0.005\text{ms}$, the solution v_0 was defined as the closest approximation to the analytical solution. We define a discrete function

$$\epsilon(i) := \|v_0 - v_i\|_{L^2(\Omega)}.$$

Here v_i are solutions to the Equation(2.1) with increased Δt for each i. The norm used is the L^2 norm, which will be discussed in more detail in 3.3.

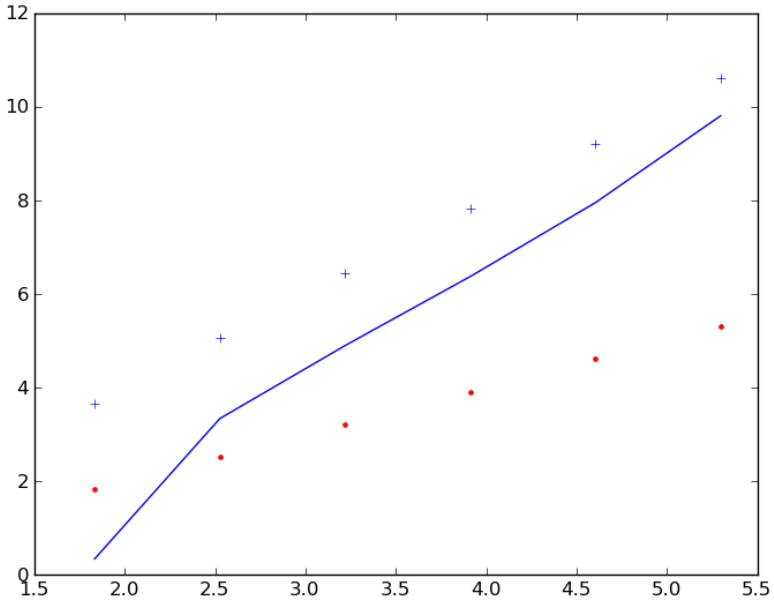

Figure 3.6: The *a postriori* error estimate. The solid line shows the error $\epsilon(i)$ plotted on a logarithmic scale, while $\cdot$ shows $\mathcal{O}(\Delta t)$ and $+$ shows $\mathcal{O}(\Delta t^2)$.

From Figure 3.6 we can see that for a large Δt the error estimate is by no means $\mathcal{O}(\Delta t^2)$. However, as Δt decreases, $\epsilon(\Delta t)$ slowly converges to $\mathcal{O}(\Delta t^2)$, which was to be expected.

3.3 Discrete norms

The norms given in Section 2.3, are applied to continuous functions. However, the solution to the monodomain Equation (2.1) is a discrete approximation. With the finite difference scheme the domain was discretized into smaller rectangular domains, $\omega_{i,j}$. Treating the solution v as constant over each of these $\omega_{i,j}$, we approximate the integral of v by:

$$\int_\Omega |f| \, d\mathbf{x} = \sum_{i=1}^{m} \sum_{j=1}^{m} \int_{\omega_{i,j}} |f| \, d\mathbf{x} = \sum_{i=1}^{m} \sum_{j=1}^{m} h^2 |f_{i,j}|.$$

This results in the following discrete norms

$$\|f\|_{L^2(\Omega)} = h \left(\sum_{i=1}^{m} \sum_{j=1}^{m} |f_{i,j}|^2 \right)^{\frac{1}{2}},$$

$$\|f\|_{H^1(\Omega)} = h \left(\sum_{i=1}^{m} \sum_{j=1}^{m} (|f_{i,j}|^2 + |\nabla f_{i,j}|^2) \right)^{\frac{1}{2}},$$

$$\|f\|_{H^2(\Omega)} = h \left(\sum_{i=1}^{m} \sum_{j=1}^{m} (|f_{i,j}|^2 + |\nabla f_{i,j}|^2 + |\nabla^2 f_{i,j}|^2) \right)^{\frac{1}{2}}.$$

The derivatives can be approximated by considering the following Taylor series:

$$f(x + \Delta x) = f(x) + \Delta x \frac{\partial f}{\partial x} + \mathcal{O}(\Delta x^2),$$

and

$$f(x - \Delta x) = f(x) - \Delta x \frac{\partial f}{\partial x} + \mathcal{O}(\Delta x^2).$$

Hence, by much the same argument as in Section 3.1.3, we achieve the following approximation

$$\frac{\partial f}{\partial x} = \frac{f_{i-1,j} - f_{i+1,j}}{2h},$$

and similarly,

$$\frac{\partial f}{\partial y} = \frac{f_{i,j-1} - f_{i,j+1}}{2h}.$$

This gives the derivative of v across the neighboring $\omega_{i,j}$ in x and y direction. For more information on implementation of norms, see Appendix B.1. When applying total variation to discrete functions, we use the same estimates on the integrals and derivatives as for the norms.

Chapter 4

Fibrillating Patterns and Norm Comparisons

4.1 Action Potential Simulator

So far, we have looked at some well-known mathematical models, describing the biophysiological properties of wave propagation in cardiac tissue. Furthermore, we discussed the bidomain model which divided the tissue into two domains, an intracellular and an extracellular domains. The domains had different anisotropic rates, in the sense that the conductivity of the tissue is directionally dependent on the intra- and extracellular domain. But by assuming equal anisotropic rates, we reduced the bidomain model into much simpler the monodomain model

$$\frac{\lambda}{1+\lambda}\nabla \cdot (M_i \nabla v) = \chi I_m. \tag{4.1}$$

Here λ is some scaler, such that the conductivity tensors $M_e = \lambda M_i$. Typical conductive values for cardiac tissue is $\sigma_l^e = 2.0\text{mS/cm}, \sigma_t^e = 1.65m\text{mS/cm}, \sigma_l^i = 3.0\text{mS/cm}$ and $\sigma_t^i = 1.0\text{mS/cm}$ [5]. It is easy to see that there exists no single λ where $M_e = \lambda M_i$. We can simplify yet again by assuming

$$\lambda = \frac{\sigma_l^e/\sigma_l^i + \sigma_t^e/\sigma_t^i}{2} \approx 1.15.$$

The parameter χ in (4.1) is the surface to volume ratio of the cell membrane, which is given as 2000cm^{-1}, see [5]. The transmembrane current

density, given by the parameter I_m, is defined by

$$I_m = C_m \frac{\partial v}{\partial t} + I_{ion}.$$

Here C_m describes electrical capacitance of the tissue, which in the physiological case is $1\mu\text{F/cm}^2$, [5]. However, this parameter will be adjusted, since the domain is unitary:

$$\Omega = \{(x, y) \in \mathbb{R} \, : \, 0 \leq x, y \leq 1\},$$

for a more realistic action potential dynamic. I_{ion} is the ionic model, which in our case is the Aliev Panfilov two variable model

$$\begin{aligned} \frac{\partial v}{\partial t} &= -(v_{rest} - v_{peak})(kV(V-a)(V-1) + Vw) + i_s, \\ \frac{\partial s}{\partial t} &= 0.25\epsilon(v, s)(-w - kV(V-a-1)), \end{aligned}$$

where

$$\begin{aligned} V &= \frac{v - v_{rest}}{v_{peak} - v_{rest}}, \\ \epsilon(v, s) &= \epsilon + \mu_1 \frac{s}{V + \mu_2}, \end{aligned}$$

This model is known for generating fibrillatory patterns and thus suits our purpose perfectly. The different constants are dependent on cell types, and setting $k = 8, a = 0.01$ and $\epsilon = 0.01$ gives action potential similar to ventricular cells. The other parameters μ_1 and μ_2 govern the restitutional properties of the tissue. We use $\mu_2 = 0.3$, while μ_1 will be varied to produce various break-up patterns. To start, we let $\mu_1 = 0.07$. Our monodomain model with Aliev Panfilov's cellular model is simulated by using the methods Strang splitting, explicit Runge-Kutta and semi-implicit finite difference method, all explained in Section 3.2.

The stimulus $i_s(t)$ is split into two parts: $S_1(t)$ represents the stimulus applied by the sinusoidal node, which is applied to the upper part of the domain. $S_2(t)$ is applied to the left side of the domain, and reflects the

distortion signal initiating fibrillation. $i_s(t)$ is then defined as

$$i_s(t) := S_1(t) + S_2(t)$$

where $S_1(t)$ and $S_2(t)$ enters 1/4 of the domain in respectable directions.

With the stimulus $i_s(t) = (15\text{mV/ms}, 0)$ on the interval $t \in [0\text{ms}, 3\text{ms}]$ we were able to induce action potential on the full domain. By regulating the electrical capacitance of the tissue, C_m, we modify the speed at which the action potential propagates through the tissue. We found that for $C_m = 1\mu\text{F/cm}^2$ the whole domain became depolarized within 20ms, while for $C_m = 15\mu\text{F/cm}^2$, the depolarization of the complete domain took 70ms, which is much closer to the physiological phenomenon. Hence, we let $C_m = 15\mu\text{F/cm}^2$.

Included below, are some figures depicting the transmembrane potential on a 200×200-mesh at different stages of a normal heart beat. In Figure 4.1 we see the rapid depolarization process spreading through the cardiac tissue. After the depolarization phase, the plateau phase follows, where transmembrane potential is hovering around the peak value. Afterwards, the repolarization of the complete tissue starts, which is depicted in Figure 4.2. We can see that the repolarization resembles a continuous trail slowly lowering the transmembrane potential to its normal resting value. During this repolarizing process, we will in the following section apply S_2 stimulus to initiate fibrillations.

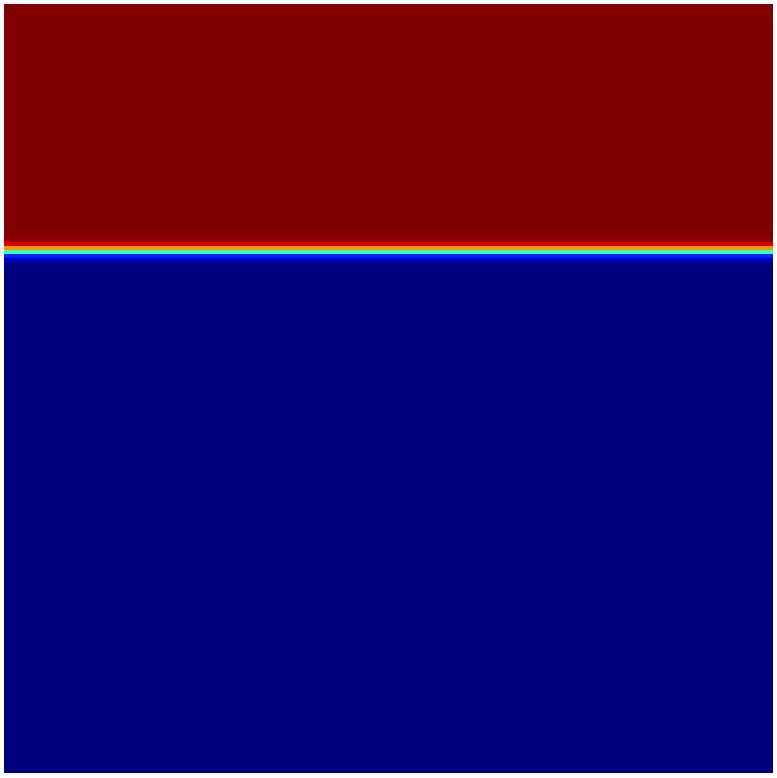

Figure 4.1: Action potential propagation after 10ms of simulation. The red represents depolarized tissue, while blue represents tissue in a resting state. In between, we see tissue that have started the depolarization process, and we see that the process is very rapid.

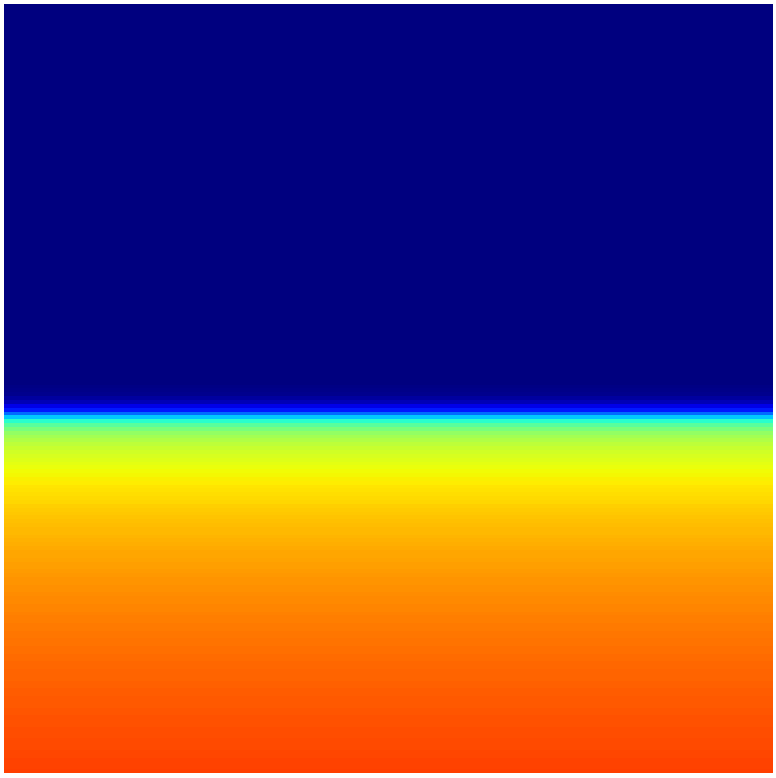

Figure 4.2: Action potential propagation after 230ms of simulation. The blue part is repolarized tissue. We see that the repolarization process is much more gradual than the depolarization process.

4.2 Fibrillating Patterns

In this section we will examine how changes to the restitutional properties of the recovery potential results in different fibrillatory patterns. We will be looking at different fibrillation patterns, from seemingly chaotic to quite stable spiral patterns. The critical parameter making the solution stable or non-stable is $\mu_1.\mu_1$ greatly affects the duration of the action potential, as shown in Figure 4.3. The figure depicts the action potential of a single cell for simulations with various μ_1-values.

We will have to apply S_2 stimulus at different times t to be able to obtain fibrillation for each simulation. By applying the S_2-stimulus at times when half of the domain is in resting state and the other half is in the repolarization phase, we can to initiate re-entry.

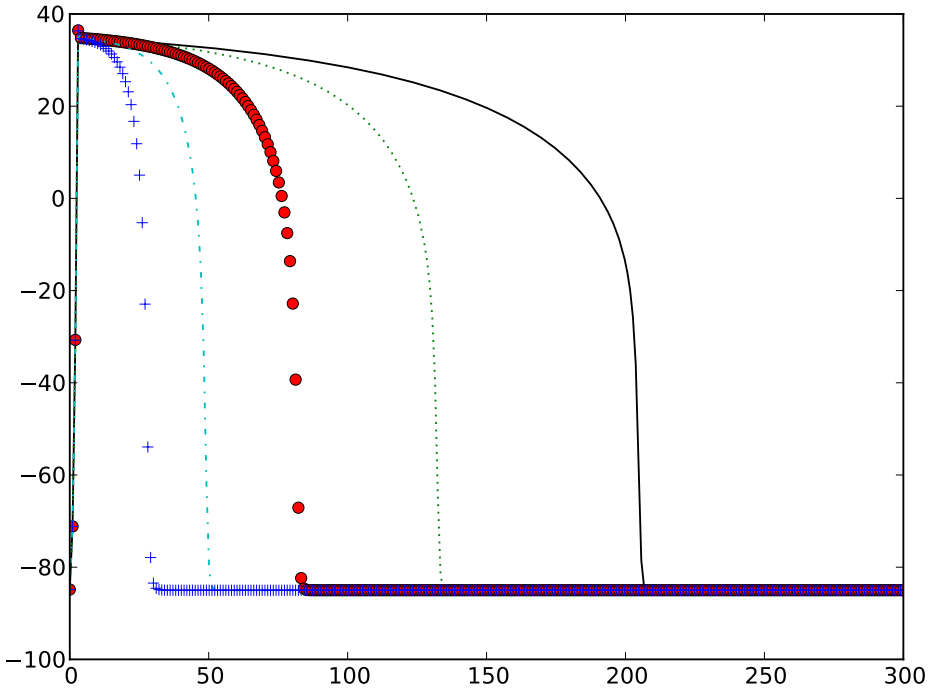

Figure 4.3: Action potential of a single cardiac cell with different values for μ_1. The x-axis shows the time, while the y-axis gives the transmembrane potential. For plot denotation see Table 4.1.

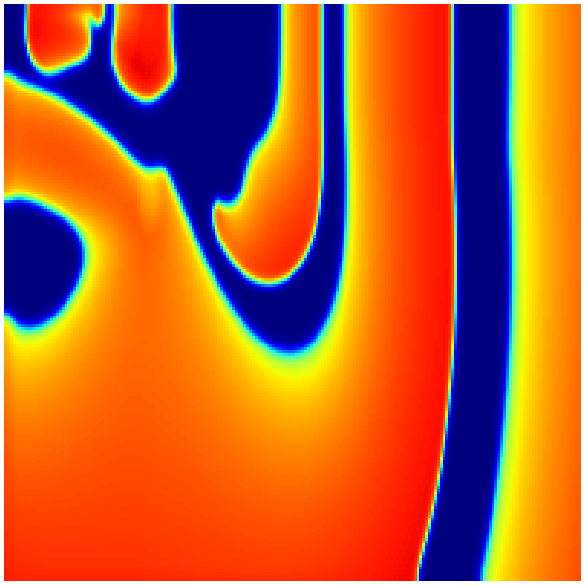

Figure 4.4: Action potential development for $t = 500$ms and $\mu_1 = 0.07$.

For each simulation a 15mV/ms, S_1-stimulus was initiated at $t = 0$, lasting for 3ms. The transmembrane potential at $t = 0$ is at resting value, i.e. $v = v_{rest} = -85$mV, while the recovery potential is $w_0 = 0$. For each simulation, we apply an $S2$-stimulus of 30mV/ms also lasting for 3ms, at a later point in time, typically in the interval $[50\text{ms}, 300\text{ms}]$, depending on μ_1. Each simulation models the development of the transmembrane potential on the interval $t \in [0\text{ms}, 1000\text{ms}]$. For each new simulation, the control parameter μ_1 is doubled until the solutions becomes non-chaotic.

Figures 4.4 and 4.5 display the behavioral pattern of fibrillatory action potential when $\mu_1 = 0.07$. In this simulation the S_2-stimulus was applied at $t = 220$ms.

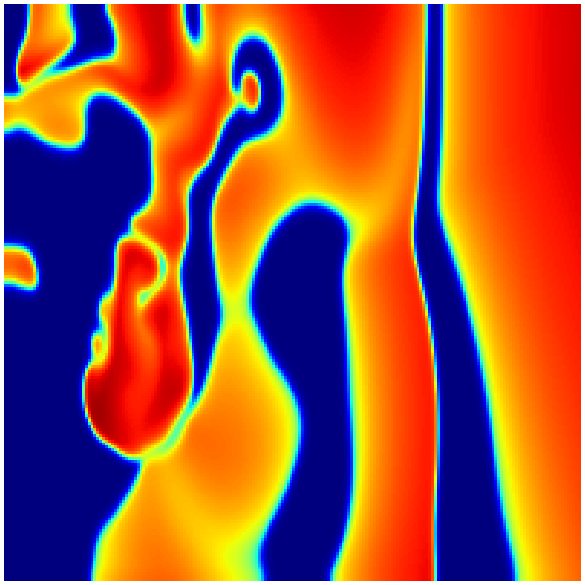

Figure 4.5: Action potential development for $t = 1000$ms and $\mu_1 = 0.07$.

We see that with $\mu_1 = 0.07$, the behavioral pattern of the transmembrane potential is chaotic. By first simulating the transmembrane potential for $\mu_1 = 0.14$ and $S_2 = 0$, we observed that half the tissue was in a repolarization phase at $t = 150$ms. Hence, using the given values for μ_1, and further inducing the S_2 stimulus at $t = 150$ms, we achieve the fibrillatory behavior, as seen in Figures 4.6 - 4.7.

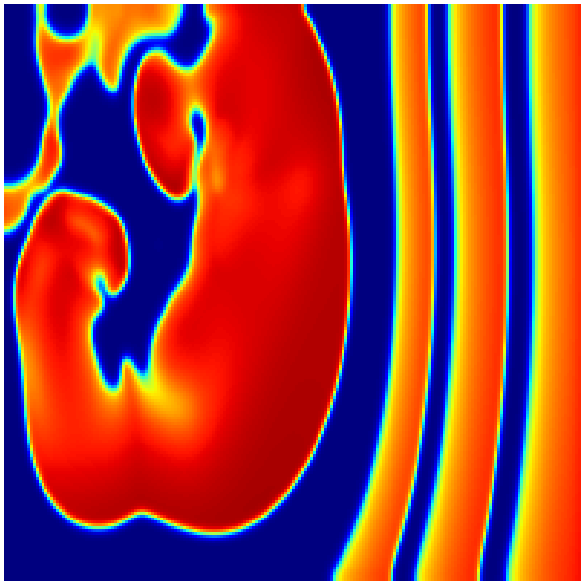

Figure 4.6: Action potential development for $t = 500$ms and $\mu_1 = 0.14$.

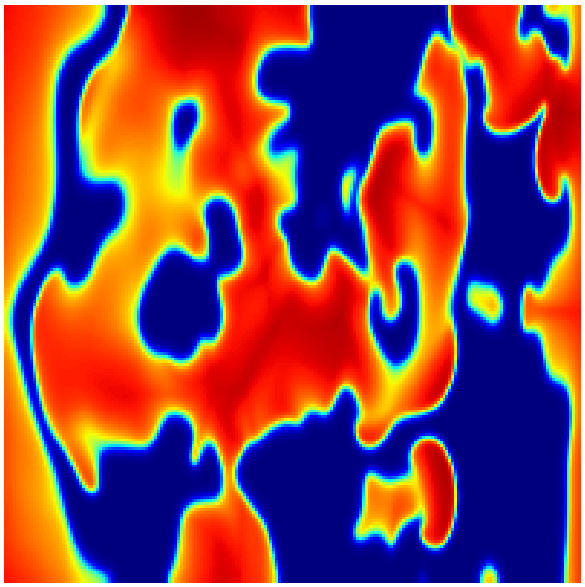

Figure 4.7: Action potential development for $t = 1000$ms and $\mu_1 = 0.14$.

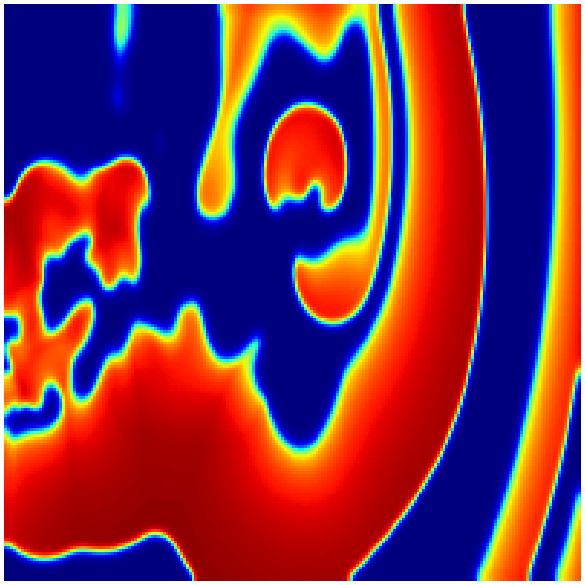

Figure 4.8: Action potential development for $t = 500$ms and $\mu_1 = 0.28$.

We observe similar chaotic behavior as for $\mu_1 = 0.07$. In the third simulation, we let $\mu_1 = 0.28$, as depicted in Figures 4.8 -4.9 . Here the S_2-stimulus is applied at $t = 100$ms, where t was found as earlier by observing the non-fibrillatory solution. With μ_1 this large, the repolarization phase is shortened, and the transmembrane potential of the upper part of the domain enters resting state before the lower part become depolarized, therefor such an early S_2 stimulus.

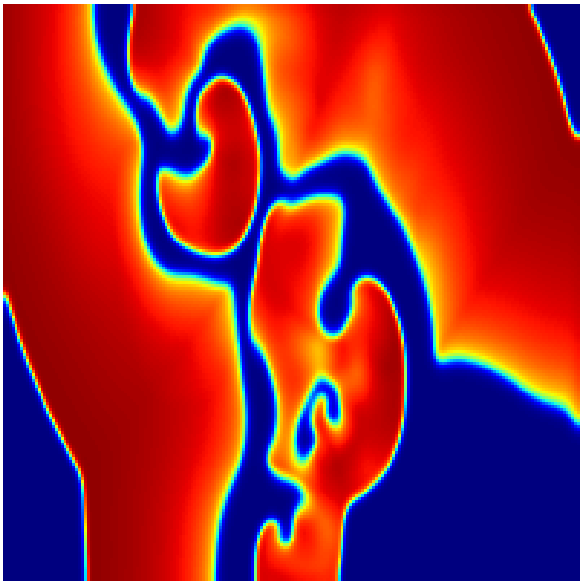

Figure 4.9: Action potential development for $t = 1000$ms and $\mu_1 = 0.28$.

When we let $\mu_1 = 0.56$, we get a very different result. At this point the repolarization phase is so short that depolarization of the lower part of the domain has not even begun before the upper part of the domain has reached its resting value. We applied the $S2$-stimulus at $t = 60$ms, and Figure 4.10 clearly shows a spiral structure. However, around $t = 800$ms, the structure breaks up and becomes chaotic as well, as shown in Figures 4.11 - 4.12. Lastly, with $\mu_1 = 1.12$ the simulation result in the stable spiral depicted in Figure 4.13. Each simulation described in this section, has begun with spiral like structures. For solutions with $\mu_1 < 0.56$, the spiral wave breaks up, as the repolarization phase is long, and thus the depolarization process catches up with the repolarization of the spiral arms, and creates an unstructured pattern. For the solution where $\mu_1 = 0.56$, the repolarization of the tissue is almost fast enough to let the transmembrane potential reach its resting state before becoming depolarized again, but after a while the solution does slowly become chaotic and unstructured. When $\mu_1 = 1.12$ the repolarization phase has been sufficiently shortened, such that the membrane potential has reached its resting value before the depolarization process of the spiral arm starts, and thus we get a stable spiral pattern.

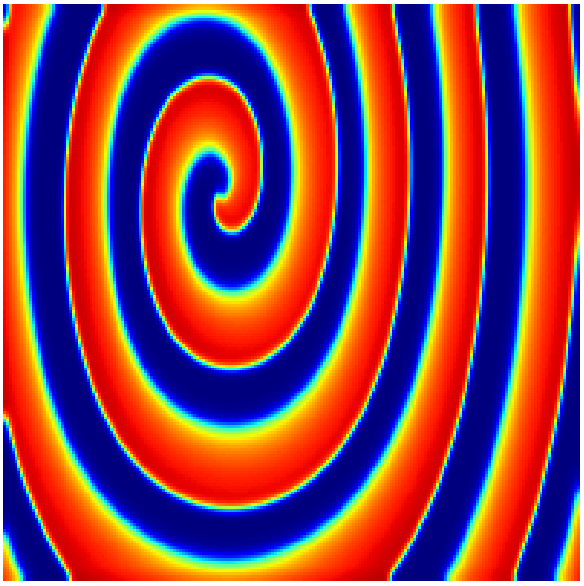

Figure 4.10: Action potential development for $t = 500$ms and $\mu_1 = 0.56$.

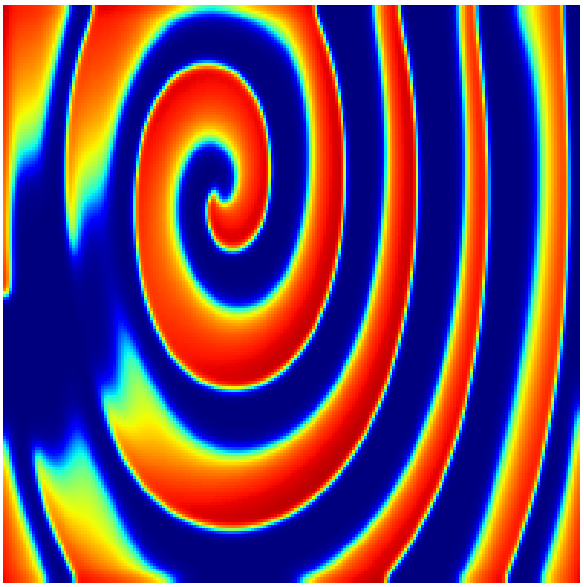

Figure 4.11: Action potential development for $t = 800$ms and $\mu_1 = 0.56$.

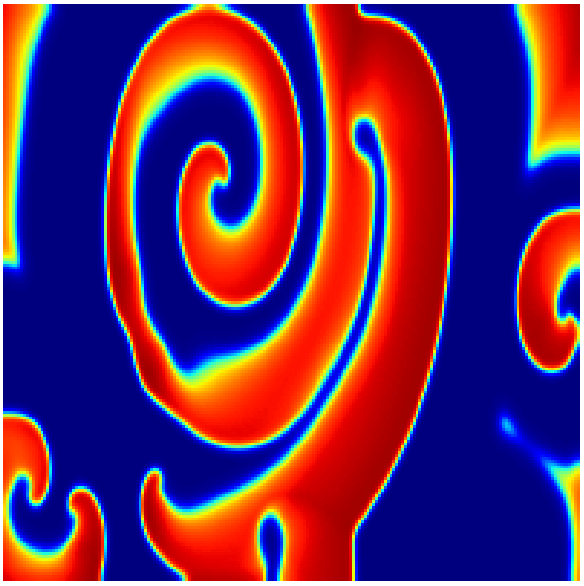

Figure 4.12: Action potential development for $t = 1000$ms and $\mu_1 = 0.56$.

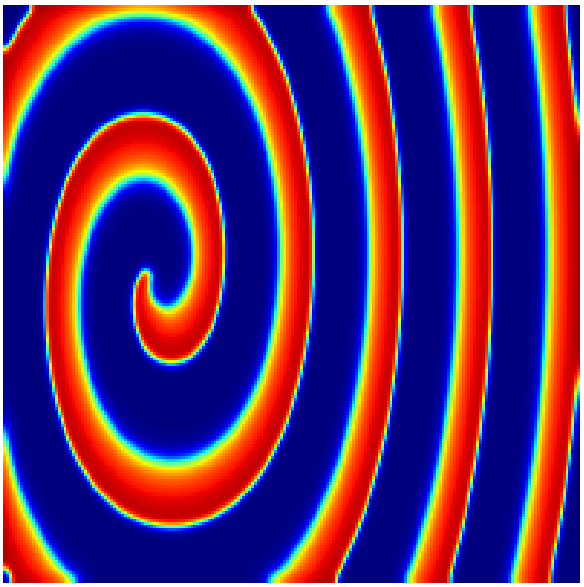

Figure 4.13: Her we see the action potential development $t = 1000$ms, with $\mu_1 = 1.12$

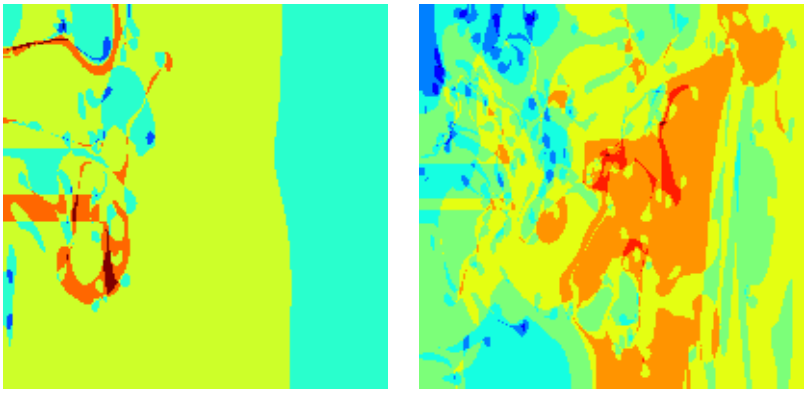

Figure 4.14: Structure of fibrillatory patterns for $\mu_1 = 0.07$ (left), $\mu_1 = 0.14$, (right).

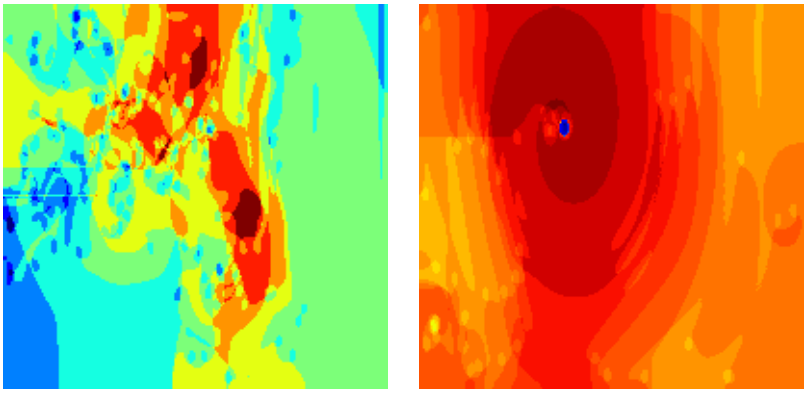

Figure 4.15: Structure of fibrillatory patterns for $mu_1 = 0.28$ (left) and $mu_1 = 0.56$ (right).

In order to get an overall picture of the behavior of the transmembrane potential in each simulation, we have constructed an algorithm recording the depolarization frequency. The algorithm simply records how often the membrane potential in a given grid point exceeds 0. The Figures 4.14 and 4.16 show the depolarization frequency of the transmembrane potential for some areas on Ω. The color scale goes from blue to red, which represents the lower and higher frequency areas of the given simulation. For more details, see Appendix B.2. For simulations with $\mu_1 < 0.56$, the pattern appear random and chaotic. Interestingly, we see that for $\mu_1 = 0.56$ and $\mu_1 = 1.12$, the frequency is more or less even across the whole tissue. Note however, that the frequency increases somewhat closer to the center of the spiral, except at the spiral tip, where the frequency is much lower. For $\mu_1 = 0.56$,

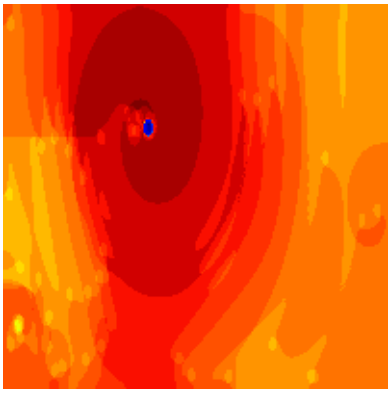

Figure 4.16: Structure of fibrillatory patterns for $mu_1 = 1.12$ (right).

μ_1	symbol	colour
0.07	$+$	blue
0.14	$\cdot$	green
0.28	$\circ$	red
0.56	$-\cdot$	cyan
1.12	$-$	black

Table 4.1: Denotation according to μ_1.

we can also see the trace meandering spiral tip caused by the re-entry of the fibrillatory signal.

4.3 Norm Comparisons

Figures 4.17-4.19 show the development of the norms (2.8)-(2.10), as described in Section 2.3, as functions of t on the intervalÂ $t \in [500\text{ms}, 1000\text{ms}]$. For t in this interval, the non-fibrillatory solutions are constant, as the transmembrane potential have reached its resting value. Thus

$$\|v_{rest}\|_{L^2(\Omega)} = v_{rest}\|1\|_{L^2(\Omega)}$$

We will concentrate on exploring features of the normed fibrillatory solution, obtained at $[500\text{ms}, 1000\text{ms}]$, as the non-fibrillatory solution is constant on the given interval.

In Figures 4.17, 4.18 and 4.19, we see the development of the L^2-, H^1- and H^2-norms respectively, for the different fibrillatory patterns obtained in Section 4.2.

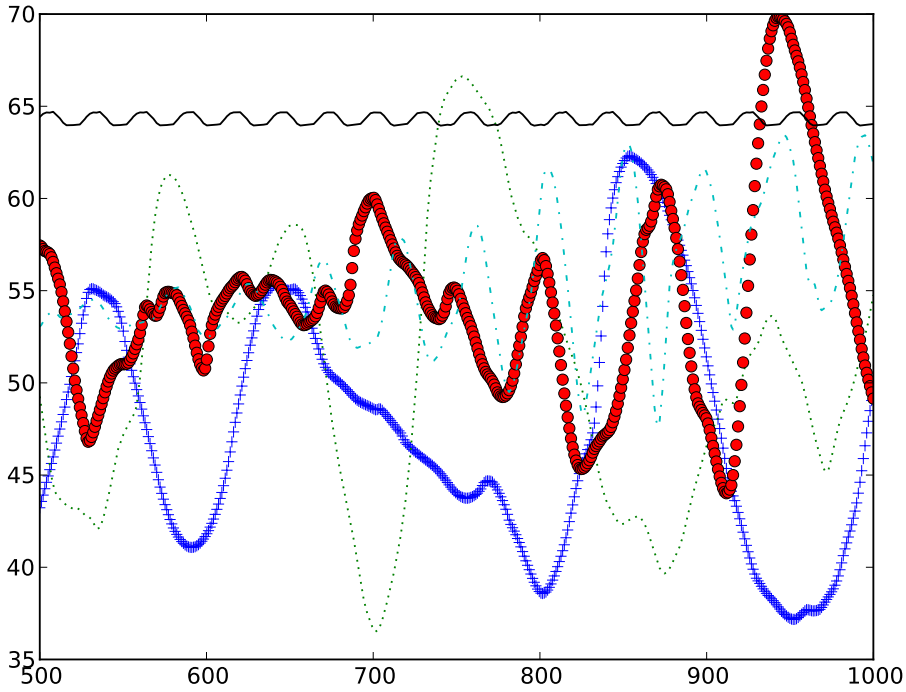

Figure 4.17: L^2-norm of the simulations run in Section 4.2. See Table 4.1 for graph denotation.

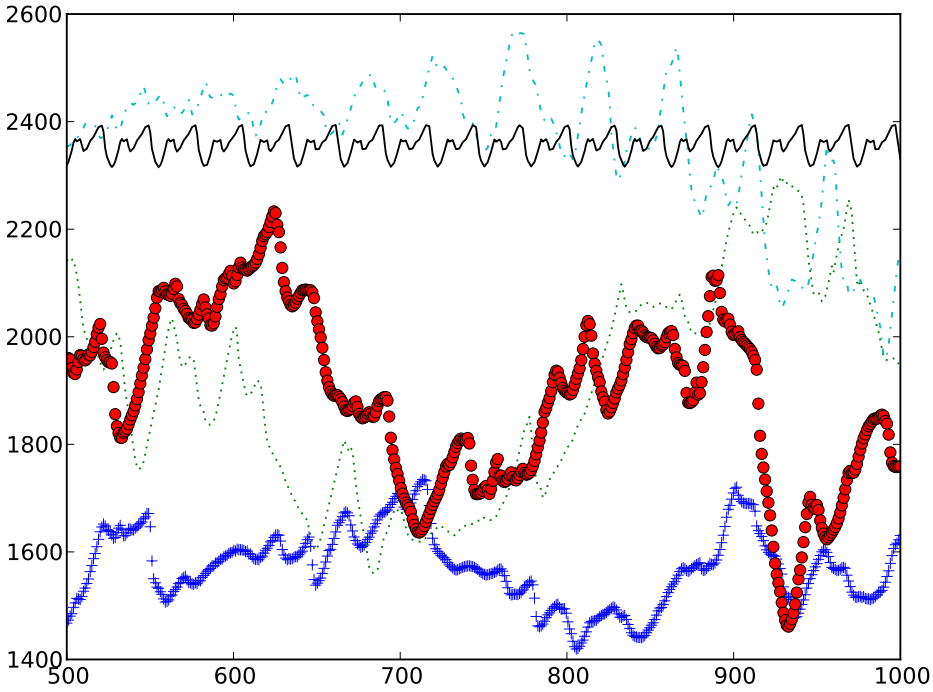

Figure 4.18: H^1-norm of the simulations run in Section 4.2. See Table 4.1 for graph denotation.

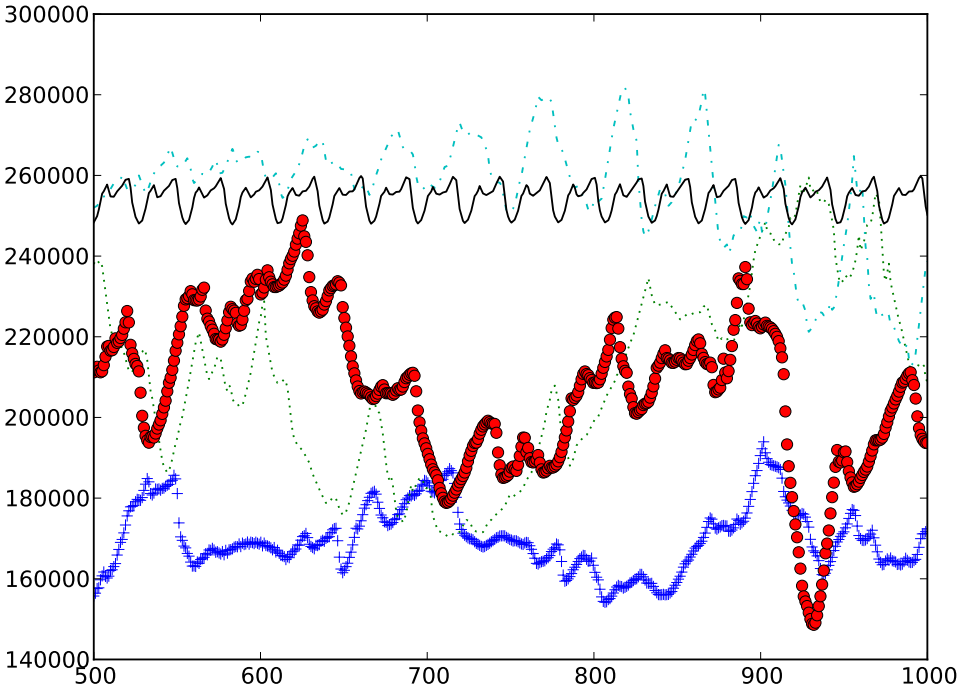

Figure 4.19: H^1-norm of the simulations run in Section 4.2. See Table 4.1 for graph denotation.

From Figures 4.17-4.19 we see that the norms on the different solutions are oscillating. However, the oscillating pattern obtained by solutions for $\mu_1 < 1.12$ are seemingly chaotic, while the pattern for $\mu_1 = 1.12$ is stable with rhythmic oscillations of minor and constant amplitude.

Figure 4.20 shows the total variation of the simulations on $t \in [500ms, 1000ms]$. We see that for the semi-norm total variation, applied to the stable spiral

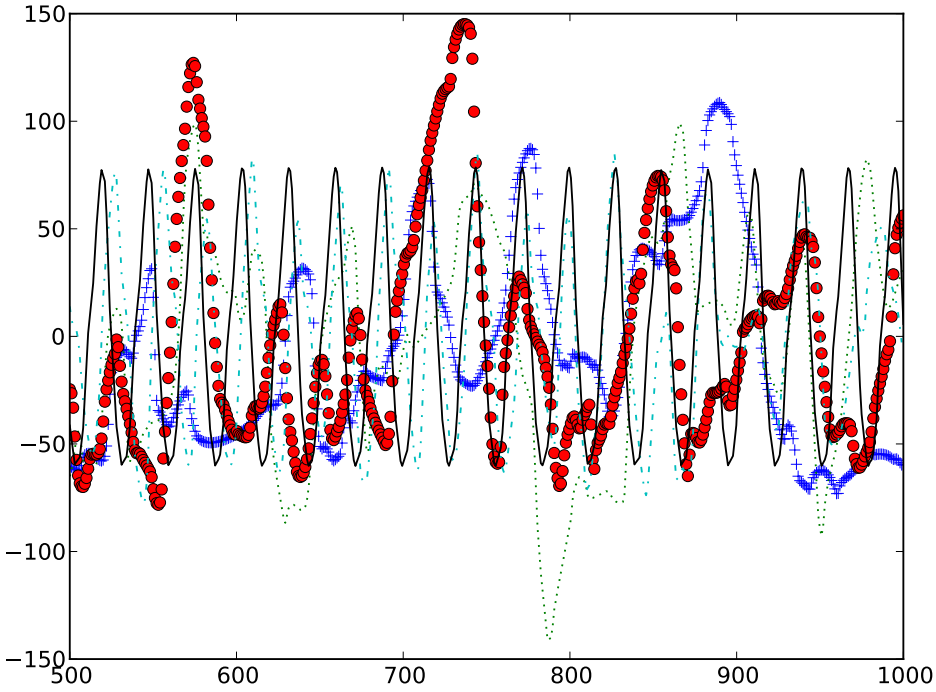

Figure 4.20: Total variation of the simulations run in Section 4.2. See Table 4.1 for graph denotation.

solution obtained by letting $\mu_1 = 1.12$, we get a periodic oscillating function with equal amplitude and constant phase. Note that the amplitude is considerably larger than for the norms. As for the norms, the behavior of the semi-norm on the other solutions are still chaotic.

An interesting feature with this norm analysis is the comparisons of solution obtained by setting $\mu_1 = 0.56$ and the stable spiral solution where

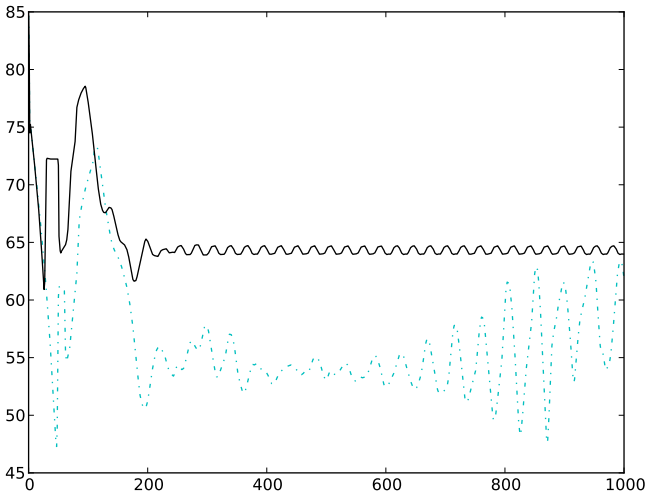

Figure 4.21: A comparisons of the L^2-norm for the solutions obtained for $\mu_1 = 0.56$ and $\mu_1 = 1.12$ on the interval $[0\text{ms}m, 1000\text{ms}]$. For graph denotation see Table 4.1.

$\mu_1 = 1.12$. The semi-stable spiral solution, where $\mu_1 = 0.56$ did not show visual signs of break up, before $t \approx 700$ where patterns started to change considerably. However, considering the the norms of this solution, espesically, the L^2-norm and the total variation semi-norm, it is clear that this result are nowhere near a periodic pattern. This is seen more clearly in the Figures 4.21 and 4.22. This property might be used to destinguish between different fibrillatory patterns.

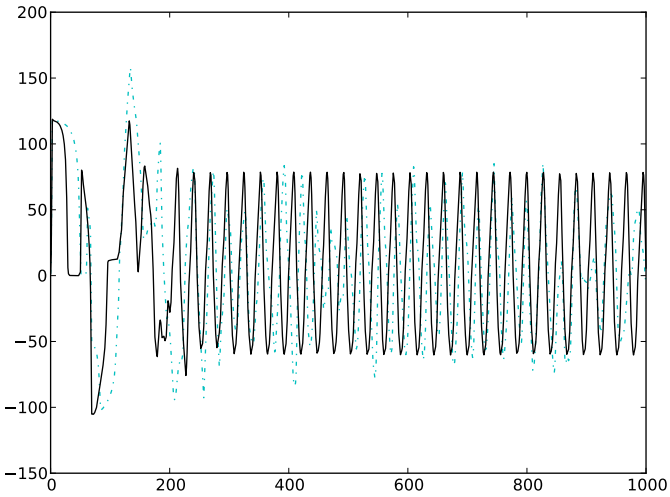

Figure 4.22: A comparisons of the total variation semi-norm for the solutions obtained for $\mu_1 = 0.56$ and $\mu_1 = 1.12$ on the interval $[0\text{ms}m, 1000\text{ms}]$. For graph denotation see Table 4.1.

Chapter 5

Conclusion

In this thesis we have looked at different mathematical models for portraying action potential propagation in the heart's excitable tissue, and we explored the different qualities of the these models in terms of reproducing this biophysiological phenomena. The models we examined were defined by a set of partial differential equations. We considered bidomain models, but as these have a high level of detail, they take too much time to solve. Thus we wanted to use a simpler model. We showed how the bidomain models can be simplified into the monodomain models, and chose the latter, as solving them are much more efficient, and we still get a sufficiently detailed model for the purpose of this thesis. The partial differential equation from the monodomain model was non-linear, and thus we chose to rather use an approximation to the solution of the monodomain equation, obtained by operator splitting. This left us with one linear partial differential equation, and an ordinary differential equation. There are many ways to model the ionic current of excitable cells, and we chose the Aliev Panfilov model, due to its ability to replicate different break up patterns.

In Chapter 3, we studied different numerical methods for solving the differential equations obtained by the monodomain model. For the partial differential equation, we first tried an explicit finite difference scheme, but as the stability requirements were strict, a very fine time resolution was necessary. The method therefore became too time inefficient. Thus we ended up using the more time-efficient semi-implicit finite difference scheme. With this solver for the partial differential equation we were able to greatly in-

crease the time steps. Thus we solved the ordinary differential equation using a simple second order explicit solver for increased accuracy.

By varying model parameters that control the recovery property of the action potential, we were able to generate different fibrillatory patterns. Further, we implemented various norms, to see if these could be used for distinguishing between laminar and turbulent flows. We applied Lebesgue- and Sobolev-norms, more precisely the L^2-, H^1- and H^2 norms, for each time step t. In addition, we also applied the total variation semi-norm. Applying these norms and the semi-norm to the solutions resulted in a discrete function $f(t)$ for each norm. We found that the function $f(t)$ was periodic for stable fibrillatory solutions for all norms and for the semi-norm, while for unstable solutions, $f(t)$ was seemingly chaotic. seemed stable early on in the simulation, but eventually became unstable after some time, $f(t)$ never converges to a periodic stable function. This leads us to believe that it might be possible to generate numerical methods for distinguishing between stable and non-stable break up patterns for the simulated solutions.

The L^2-norm and total variation semi-norm generated the easiest recognizable periodic wave. By finding whether these norms applied to a given solution generates a periodic repeating pattern or not, we might also be able to find if the solution is stable. By running it at different time intervals, this might reveal if the solution will generate a chaotic or non-chaotic structure of the transmembrane potential. It does however take some time before $f(t)$ converges to a periodic stable function, and over a short time window it is impossible to determine whether or not the solution will become fibrillatory stable or not. As seen in Figures 4.21 and 4.22, $f(t)$ was not periodic for stable solutions, before $t \approx 250$ms. One of the problems is thus to find when $f(t)$ converges, if the solution is stable.

An approach not used in this thesis is discrete cosine transforms. Discrete cosine transforms expresses a function or signal in terms of a sum of sinusoids with different frequencies and amplitudes. The transform operates on a function with finitely many data points. The most common of these transforms is the type-II DCT, some times called the DCT. In two dimensions

the DCT is given by

$$V_{k,l} = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} v_{i,j} \cos\left[\frac{\pi}{m}\left(i+\frac{1}{2}\right)l\right] \cos\left[\frac{\pi}{m}\left(j+\frac{1}{2}\right)k\right],$$

where $v_{i,j}$ is the discrete data points for $i,j = 0,\ldots,N-1$ and $l,k = 0,\ldots,m-1$. Since the DCT transforms the signal into the frequency domain one would expect to able to distinguish between the types of solutions based purely on data from a single time step. The frequency spectrum of a chaotic solution would typically be broader.

Appendix A

Implementation of Solvers

A.1 The Partial Differential Equation Solver

When we constructed the numerical model, we quickly found that one of the most tasking calculations was constructing the sparse matrices A from Section 3.2.3. However, A can be written as a scalar multiplied by matrix A_0, and this matrix depends only on the discretization of the domain. We construct A_0 in *Python* as follows:

When writing A_0 to file, we can simply load the matrix into the solver for each simulation, as long as the the spatial discretization remains the same. The full solver was constructed as an object. As input when initiating the object, we use the variables defining the discrete time and space domains. Also we imported the pre-made matrix A_0 and declared a LU-factorized solver for the matrix $I + r/2\Delta t A_0 = B$. This was done as follows:

```
self.B = self.r*self.dt*self.theta*A_0; self.B.setdiag(1 + A_0.diagonal())
A = -self.r*self.dt*self.theta*A_0; A.setdiag(1 + A_0.diagonal())
self.solver = scipy.sparse.linalg.factorized(A.tocsc()
```

Note, that this was only done once. When solving the partial differential equation, we called the method:

```
def PDEsolver(v):
  V = numpy.reshape(v, numpy.size(v))
  b = self.B.dot(V); u = self.solver(b)
  return numpy.reshape(u, numpy.shape(v))
```

```
import scipy.sparse as sp
import shelve
def stiffnessMatrix(n):
  dx = 1.0/n; sigma_l = 3.0; sigma_t = 1.0

  A = sp.lil_matrix((n**2, n**2))
  A += (- 2*sigma_l - 2*sigma_t)*sp.eye(n**2, n**2, format = 'lil') + \
    sigma_t*sp.eye(n**2, n**2, k = 1, format = 'lil') +\
    sigma_t*sp.eye(n**2, n**2, k =-1, format = 'lil') +\
    sigma_l*sp.eye(n**2, n**2, k = n, format = 'lil') +\
    sigma_l*sp.eye(n**2, n**2, k =-n, format = 'lil')

  for i in xrange(1,n):
    A[i*n, i*n+1] += sigma_t
    A[i*n, i*n-1] -= sigma_t
    A[i*n-1, i*n-2] += sigma_t
    A[i*n-1, i*n] -= sigma_t
  A[0,1] +=  sigma_t
  A[n**2-1, n**2-2] += sigma_t
  A[:n,:] += sigma_l*sp.eye(n, n**2, k = n, format = 'lil')
  A[n**2-n:,:] += sigma_l*sp.eye(n, n**2, k = n**2-2*n, format = 'lil')

  fil = "./data/matrix/stiffnessMatrix%.4d" % n
  save = shelve.open(fil); key = "matrix"
  save[key] = A; save.close()
```

A.2 The Ordinary Differential Equation Solver

The ordinary differential equation was solved using the the following explicit Runge-Kutta method:

```
def RKtrans(self, v, w, i_st):
  dt = self.dt*self.theta/self.m
  f1 = self.f(v, w, i_st)
  f2 = self.f(v + dt*f1, w, i_st)
  return v + dt*0.5*(f2 +f1)

def RKrecovery(self, v, w):
  dt = self.dt*self.theta/self.m
  g1 = self.g(v, w)
  g2 = self.g(v, w + dt*g1)
  return w + dt*0.5*(g2 + g1)
```

We also confirmed that our explicit Runge-Kutta method gave an accurate representation of the action potential on a single cell, by comparing the results to a much slower singly-diagonally-implicit Runge-Kutta method:

```
def J(self, v, w, i_st):
  V = (v - self.vRest)/(self.vPeak - self.vRest)
  eps = self.eps + self.mu1*w/(V+self.mu2)
  epsV = self.mu1*w*(1.0/(self.vPeak - self.vRest))/(V + self.mu2)**2
  epsS = self.mu1/(V + self.mu2)
  J11 = self.k*((V - self.a)*(V - 1) + V*(V -1) + V*(V - self.a)) + w
  J12 = V
  J21 = 0.25*epsV*(-w -self.k*V*(V - self.a -1 )) -\
     0.25*eps*(self.k*(1.0/(self.vPeak - self.vRest))*((V - self.a - 1) + V))
  J22 = 0.25*epsS*(-w -self.k*V*(V - self.a - 1)) - 0.25*eps
  return np.array([[J11, J12],[J21, J22]])

def f(self, v, w, i_st):
  V = (v - self.vRest)/(self.vPeak - self.vRest)
  return -(self.vPeak - self.vRest)*(self.k*V*(V -\
     self.a)*(V - 1.0) + V*w) + i_st
def g(self, v, w):
  V = (v - self.vRest)/(self.vPeak - self.vRest)
  eps = self.eps + self.mu1*w/(V + self.mu2)
  return 0.25*eps*(-w - self.k*V*(V - self.a - 1.0))

def SDIRK(self, v, w, i_st):
  dt = self.dt*self.theta
  for i in xrange(self.m):
    for j in xrange(self.m):
      y = np.array([v[i,j], w[i,j]])
      A =  np.eye(2,2) - dt*0.5*self.J(v[i,j],w[i,j],i_st)
      while True:
        F = - y + dt*0.5*np.array([self.f(y[0], y[1],\
            i_st[i,j]), self.g(y[0], y[1])]) +\
            np.array([v[i,j], w[i,j]])
        dy = np.linalg.solve(A, F)
        y += dy; c += 1
        if np.abs(dy.max()) < 0.001: break
      y = np.array([v[i,j], w[i,j]]) + dt*np.array([self.f(y[0],y[1],\
            i_st[i,j]), self.g(y[0],y[1])])

      v[i,j] = y[0]; w[i,j] = y[1]
```

This method gave much the similar representation as the explicit method, but was deemed to slow for practical use.

Appendix B

Implementation of Numerical Comparison Devices

B.1 Norms and Semi-Norms

The norms were calculated using the following methods:

```
def L2(func, h):
  func = numpy.reshape(func, numpy.size(func))
  norm = numpy.sum(numpy.abs(func)**2)
  return h*numpy.sqrt(norm)
```

```
def H1(func, h):
  dx = deriv_x(func,h)
  dy = deriv_y(func,h)
  return numpy.sqrt(L2(func,h)**2 +\
    L2(dx,h)**2 + L2(dy,h)**2)
```

```
def H2(func, h):
  dx = deriv_x(func,h)
  dy = deriv_y(func,h)
  dxdx = deriv_x(dx, h)
  dxdy = deriv_x(dy, h)
  dydx = deriv_y(dx, h)
  dydy = deriv_y(dy, h)

  return numpy.sqrt(L2(func,h)**2 +\
    L2(dx,h)**2 + L2(dy,h)**2 +\
    L2(dxdx, h)**2 + L2(dydy, h)**2 +\
    L2(dxdy, h)**2 + L2(dydx, h)**2)
```

```
def TV(func, h):
  TV = numpy.sum(numpy.sum(deriv_x(func,h) +\
      deriv_y(func,h)))*h**2
  return TV
```

where $h = \Delta x = \Delta y$, and $func = v$ (the transmembrane potential). The derivatives were found using the following algorithm:

```
def deriv_y(func, h):
  dy_val = (numpy.roll(func.transpose(),1) -\
    numpy.roll(func.transpose(), -1))/(2*h)

  dy = numpy.zeros(numpy.shape(func))
  dy[1:len(func)-2, 1:len(func)-2] += \
    dy_val[1:len(func)-2, 1:len(func)-2]

  return dy
```

and

```
def deriv_x(func, h):
  dx_val = (numpy.roll(func,1) - numpy.roll(func,-1))/(2*h)

  dx = numpy.zeros(numpy.shape(func))
  dx[1:len(func)-2, 1:len(func)-2] += \
    dx_val[1:len(func)-2, 1:len(func)-2]

  return dx
```

B.2 Frequency Measurement

```
def frequency(data, name):
  measure = data[0]
  compare = np.zeros(np.shape(measure))
  cc = np.zeros(np.shape(measure))
  for i in xrange(len(data)):
    func = data[i]
    for k in xrange(len(measure)):
      for l in xrange(len(measure)):
        if func[k,l] > 0.00 and cc[k,l] == 0:
          compare[k,l] += 1
          cc[k,l] = 1
        elif func[k,l] <= 0 and cc[k,l] == 1:
          cc[k,l] = 0
  fname = "./data/svg/%s.png" % name; typ = "png"
  p.imsave(fname, compare, format = typ)
```

This method gave the depolarization frequency for each grid point in the domain Ω. The color scheme of each plot is relative to each simulation.

Bibliography

[1] R. Winther A. Tveito. Introduction to partial differential equations. *A computational approach*, 1998.

[2] Rubin R. Aliev and Alexander V. Panfilov. A simple two-variable model of cardiac excitation. *Chaos, Solitons and Fractals*, 7(3):293 – 301, 1996.

[3] Leslaw K. Bieniasz, Ole Østerby, and Dieter Britz. The effect of the discretization of the mixed boundary conditions on the numerical stability of the crank-nicolson algorithm of electrochemical kinetic simulations. *Computers and Chemistry*, 21(6):391 – 401, 1997.

[4] N. Hooke, C.S. Henriquez, P. Lanzkron, and D. Rose. Linear algebraic transformations of the bidomain equations: Implications for numerical methods. *Mathematical Biosciences*, 120(2):127 – 145, 1994.

[5] A. Tveito X. Cai K. A. Mardal J. Sundnes, G. T. Lines. Computing the electrical activity in the heart. *Monographs in computational science and engineering*, 2006.

[6] B. P. Rynne and M. A. Youngson. Linear functional analysisi. pages 1 – 82, 2008.

[7] Kirill Skouibine, Natalia Trayanova, and Peter Moore. A numerically efficient model for simulation of defibrillation in an active bidomain sheet of myocardium. *Mathematical Biosciences*, 166(1):85 – 100, 2000.

[8] J. Sundnes, G. T. Lines, and A. Tveito. An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso. *Mathematical Biosciences*, 194(2):233 – 248, 2005.

[9] J. W. Thomas. The effect of the discretization of the mixed boundary conditions on the numerical stability of the crank-nicolson algorithm of

electrochemical kinetic simulations. *Texts in Applied Mathematics*, 22, 1995.

[10] Miguel Valderrábano. Influence of anisotropic conduction properties in the propagation of the cardiac action potential. *Progress in Biophysics and Molecular Biology*, 94(1-2):144 – 168, 2007. Gap junction channels: from protein genes to diseases.

[11] Liping Wen, Yuexin Yu, and Shoufu Li. Stability of explicit and diagonal implicit runge-kutta methods for nonlinear volterra functional differential equations in banach spaces. *Applied Mathematics and Computation*, 183(1):68 – 78, 2006.