Adaptive Multigrid Solvers for Space-Time Discretisations

Lisa Gaedke-Merzhäuser

November 26, 2018

Contents

Chapter 1

Prologue

Find a way to go from very broad title to heart ....

I okay in Prologue / Epilogue ?

structure of the thesis
At the beginning of each chapter I will give an overview of its content and purpose as well as intuitive explanations of the core concepts and ideas it entails.

specific versus how this can be generalised ...

non exhaustive overview cardiac electrophysiology, more details can be found ...

Fahrplan – how to tackle this

- We reformulate it as a mimimisation problem J whose solution coincides with the one of the original equation.
- Then we use a Newton iteration of the following form to solve this non-linear functional — where do we linearise ...?
- That is we set up a linearised space-time Least Square Formulation in each iteration which gives rise to system of equations of the form $Au = f$.
- And then solve this using a multigrid method.
- we get an updated solution for our Newton iteration and can repeat the process

In order to obtain a meaningful solution for u we need a number of properties to be fulfilled. In each Newton step the multigrid solver has to converge to the solution of the linearised least squares minimisation problem which mimics the corresponding linearisation of the original PDE. In the outer iteration we need the Newton method to converge to the minimum of our non-linear functional whose solution as mentioned above is supposed to correspond to the solution of the original problem.
Our hearts are absolutely vital for our survival. While it normally functions with an incredible reliability and accuracy that does not even let us begin comprehend the complexity of the

mechanisms involved, cardiovascular diseases are estimated to make up for more than 30% of all world wide's death [1]. Often this is related to abnormal heart contractions and thus understanding the processes involved in governing our heart beats is crucial to explain heart failues.
Traditionally partially differential equations are solved by recursively computing an approximation for all space-elements or nodes at a certain time t_n and then using those results to compute the approximate solution at the next time step t_{n+1}. This seems like the natural way to perform operations, for one because this is how move through time, sequentially, and second because for any real life system the previous time steps often give a good approximation for the next one, so why not use that information. Computationally this has a large drawback
In times where we can no longer (easily or significantly) speed up sequential operations but can only increase the number of parallel tasks being
Picture: what does wavefront look like?
As we can see there are two (almost) constant regions, the activated and not-activated areas. Wavefront where things are happening.
Would like to find a way to take this into account so that less computational resources are lost on the constant areas that but maintaining a high resolution at the wavefront.

The aim is to achieve this through an adapted algebraic multigrid formulation (rule, ...?) that takes the specific behavior into account, that is

The reason for us to consider a least squares formulation is due to the fact that the differential operator arising through the partial derivative in time is not symmetric. This can be seen by looking at the asymmetry of the primary variatinal formulation of the original problem.

This would not be an issue if we were to solve the equation sequentially since our set up would then be a different one, where the differential operator would only consist of the diffusion term in space. However in the space-time setting that will be introduced in more detail later the first order derivative is part of the "differential FEM operator". By using a least squares finite element formulation we give rise to a symmetric system. How this is achieved will be explained in the follwoing sections.
this thesis tries to tie together a variety of ideas to obtain an efficient solver for ... Attempting to make use to the favourable attributes of each i... while trying to avoid the pitfalls.

We would like to solve a parabolic reaction diffusion equation with a nonlinear forcing term, examples where these occur. here we focus on electrophysiology.
The main ingredients are Newton iteration, formulate problem as two coupled first order system which are turned into a space-time lsfem system using a galerkin (?) approach? in each iteration we solve a linearisation of the the problem using a multigrid method with a ... smoother and a (something about coarsening strategy)

therefore for the reader not to get lost in one of the many intermediate steps leading to our overall set up I (is that okay?) will try also keep reminding ourselves where we are in the bigger picture in each individual section.
in the following chapter each of the individual puzzle pieces will be briefly introduced, important ideas, underlying concepts or different ways of how they work. Then we will slowly be starting to put the pieces together. That is forming a space-time least squares element setting for a non-linear equation which is then put into a Newton iteration where in each step a multigrid solve is required.
Short intro: Since we are dealing with a non-linear equation we cannot simply try to find to set up a linear it is necesary to employ . The usual approach is to try to set up a set up system of equations,

Chapter 2

Cardiac Electrophysiology

The heart acts as a double pump made out of muscle tissue that provides our bodies with freshly oxygenated blood [2]. A heart has about the size of a fist and sits between our two lungs. It consists of 4 chambers, the upper two atria, that is the left and right atrium and the lower two ventricles which have connections through four heart valves that can open and close respectively but only allow the blood to flow in one direction. The left and the right side of the heart is separated by a wall of tissue known as the atrioventicular septum. An intricate interplay of contraction and relaxation of the chambers governed through electrical stimuli lead to a stable blood flow that enables the replenishment of oxygen levels of the cells in our body. Below we can see a schematic image of the heart, where the arrows indicate the direction of blood flow.

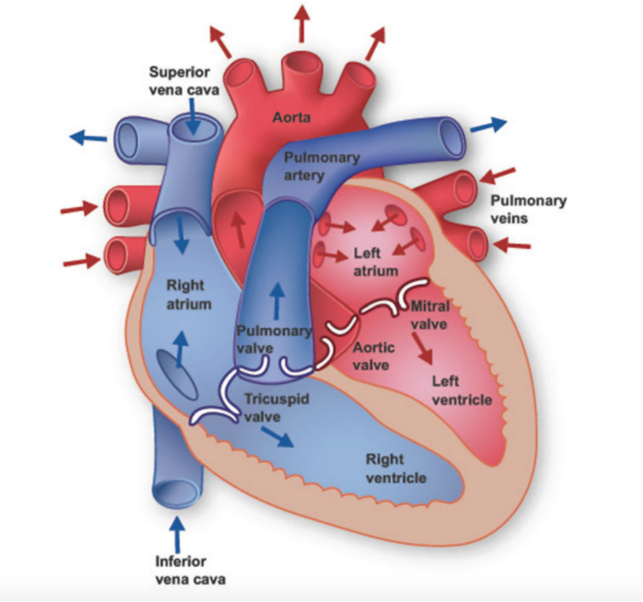


Figure 2.1: Scheme of the Heart [3]

The blood flow through the different chambers of the heart occurs in repeating cycles. After circulating through the body low oxygenated blood flows back into the heart through our veins and enters into the right atrium, which contracts once it is full. This contraction causes a pressure built up and pushes the tricuspid valve open. The blood rushes into the right ventricle, whose walls, once filled, also begin to contract, the pressure within rises again, which shuts tricuspid valve and opens the pulmonary valve to the pulmonary artery from where the blood reaches the lungs and replenishes it oxygen stocks. Afterwards it returns to the left side of

the heart from the pulmonary veins to the left atrium, which again, once it is completely filled, contracts and hereby opens the miral valve and forces the blood into the left ventricle. The left ventricle then pumps the oxygenated blood through the aortic valve into the aorta from where it flows into different parts in the body to supply cells with oxygen and nutrients before returning to the right atrium and repeating its cycle. The mitral and tricuspid valves open, and the aortic and pulmonic vavles close while the ventricles fill with blood. In contrast the mitral and tricuspid valves shut, and the aortic and pulmonic valves open during ventricular contraction. This particular sequence makes sure that all ventricles are filled up to capacity before pumping and that blood flows only in one direction. For references and further information see [4], while the following sections are based on [3].

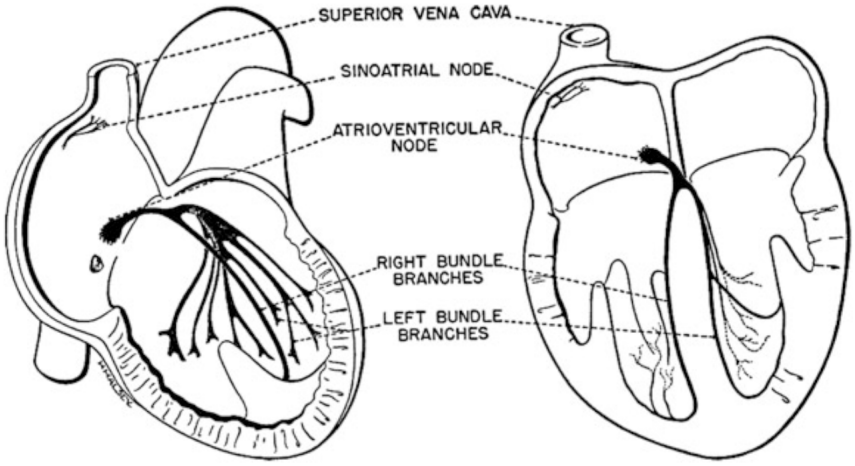


Figure 2.2: Overview of the Hearts Conduction System [3]

The heart contractions are initiated by an electric activiation, that is a depolarizing transistory membrane current which raises the transmembrane potential from its resting value of about -90 to -80 mV to small positive values. This potential describes the difference in the electric potential between the interior and exterior of the cell. The increase is followed by a repolarization current which sends the transmembrane potential back to its resting value. The initial electrical stimulus is generated by the sinoatrial node which is located on the right atrium close to the superior vena cava and possesses the ability to excite its cells autonomously. The frequency of its stimuli is dependent on the parasympathetic nervous system and hormonal factors but under normal health and stress conditions ranges from about 60-100 times per minute. The signal is then transmitted through the surrounding cells and cardiac conduction pathways to the various chambers of the heart. It first propagates to the right atrium and through Bachmann's bundle to the left atrium where it stimulates the cardiac muscle cells of the atria to contract. The activation front then travels to the atrioventricular node situated at the base of the atria. The cells there have a relatively slow conduction velocity and therefore cause a delay in the transmission which is timed this way to achieve optimal pump activity. From the atrioventricular node the stimulus reaches specialised fibres in the bundle of His and the Purkinje network that branch in the left and right bundle onto the inner surface of the ventricals. Again causing a contraction of the cardiac muscle tissue.

2.1 Activity on the Cellular Level

The heart's walls can be subdivided into 3 different layers; the inner endocardium which surrounds the the heart chambers; the outer endocardium which protects and delimits the heart from other parts of the body and the predominant middle layer consisting of cardiac muscle tissue called myocardium. This is where the conduction of the electric potential and the heart contractions mainly take place. Myocardium is made up of sheets of cells, where each one is roughly of a cylindrical shape with a size ranging from 100-150 μm by 30-40 μm. (diff source 50-150, 10-20) They are organised in a way similar to a brick wall and joined together at the ends by intercalated disks turning them into long fibres. The disks allow for easy ion movement between the cells and thus allowing for a rapid transmission of electrical impulses. Each cardiomyocyte that is each cardiac muscle cell contains bundles of myofibrils which are protein fibres which can slide past each other, making it possible for the tissue to contract. The cell's membrane is called sarcolemma and contains certain transmembrane channels whose opening and closing is governed through electric stimuli (mainly transversal cell direction?). The intercalated disks allow the transit of ions through channels called gap junctions which are predominantly in longitudinal fiber direction. Due to the varying density of the gap junctions in the different directions there is an anisotropic propagation of the electric potential throughout the tissue which complicates its simulation.

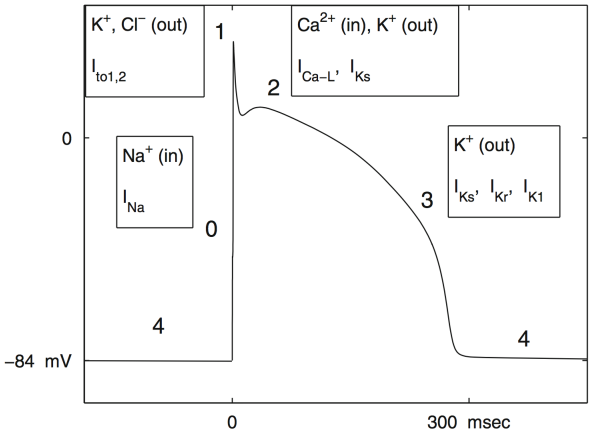


Figure 2.3: Different Phases of Cardiac Action Potential, from Franzone book

In this figure we can see a standard ventricular action potential in its main phases, that is the electric charge distribution a cell goes through over time. Following [3] we will have a brief look at the different stages that occur and what they entail.

Phase 0: Depolarization of the cell by opening of Na^+ channels of the sarcolemma which leads to rapid inflow of Na^+ ions into the cell. Hence, the transmembrance potential passes from negative to positive values.
Phase 1: Outward flow of K^+ and Cl^- ions after the inactivaiton of Na^+ channels, which causes a rapid decrease of the potential
Phase 2: Governed by an inward as well as an outward current of Ca^{2+} and K^+ ions respectively such that there almost is a balance in the potential
Phase 3: Repolarization of the cell by closing of the Ca^{2+} channels while outward current of K^+ ions is maintained therefore returning the potential to negative values.
Phase 4: The potential remains at a constant negative value. Some channels are kept open to allow for keeping the right inter-and extracellular charge balance. The cardiomyocyte stays in

this resting state until the next stimulation.

The stimuli are under normal conditions about 0.6-1 seconds apart, which means that the cell is about half the time in its resting phase. They travel as a wave through the cardiac tissue, where one cell excites the next. After this brief description of the processes involved in the functioning of the human heart, let us in the subsequent section turn towards the question of how to adequately represent them in a model and what the particular difficulties are that arise.

2.2 The Monodomain Equation

The propagation of these stimuli is usually modeled with either a version of the bidomain equation or monodomain equation. The former was first developed in the late 1970's and is the more comprehensive one of the two still posing many computational challenges in its implementation and execution. Therefore one often relies on a simpler monodomain model which in the vast majority of applications leads to a very similar solution and can therefore be considered as an adequate approximation, [5]. The discrete cellular structure is replaced by an averaged continuous model, giving rise to a parabolic reaction-diffusion equation derived from the cable equation maintaining a conservation of charge. Cable theory is used to calculate the electric current in nerve fibres by modeling them as composed segments with capacitances and resistences combined in parallel. The diffusion term represents the spread of current through gap junctions and cardiac tissue while the reaction terms describe the flux of ions across the myocyte membrane. (not exactly sure yet how things will look like, therefore the mess ...)

$$\begin{aligned} \partial_t u - \nabla \cdot (D(x)\nabla u) = I_{\text{int}}(u) + I_{\text{ext}}(x,t) \quad (x,t) \in \Omega \\ \nabla u \cdot n = \quad (x,t) \in \Gamma_N \end{aligned} \tag{2.1}$$

$u(x,t)$: electric potential
$D(x)$: conductivity tensor
$I_{\text{int}}(u) = u(u-1)(\alpha - u),\ 0 < \alpha < 1$: internal
$I_{\text{ext}}(x,t) =$ external current

There are a number of difficulties that arise when trying to numerically approximate this problem. First it is worth noting that the reaction term is non-linear and therefore requires an appropriate solver to handle this. Furthermore any heart simulation has to deal with the challenging task of dealing with large differences in spatial and temporal scaling due to the complex multiscale structure. As mentioned before the microscopic scale the electric conduction of excitation fronts happens through ion channels of cellular membranes which are on a scale of the order of 0.1 mm. On the other hand the overall size cardiac tissues invovled entails a size of several centimeters which leads to a spatial spread factor of up to 10^3. Similarly for the time parametrisation we have an even larger spread factor. A normal heartbeat takes about 1 second, that is one full cycle, whereas the step excitation front that is described in phase 0 in the previous section ranges on a much shorter time scale therefore requiring time steps within a range of about 0.01 milliseconds for accurate representations [6].

Chapter 3

Mathematical Ingredients

3.1 Space-Time Solvers

Usually the time direction in partial differential equations is not used for parallelisation. But with increasingly complex models, especially when many small steps in time are required and the rise of massively parallel computers, the idea of a parallelisation of the time axis has experienced a growing interest. Once parallelisation in space saturates it only seems natural to consider this remaining axis for parallelisation, after all time is just another dimension. However evolution over time behaves differently from the spatial dimensions, in the sense that it follows the causality principle. Meaning that the solution at later times is determined through earlier times whereas the opposite does not hold. This is not the case in the spatial domain.

The earliest papers on time parallelisation go back more than 50 years now to the 1960's, where it was mostly a theoretical consideration, before receiving an increasingly growing interest in the past two decades due to its computational need and feasibility. As mentioned in [7], on which this section is mainly based on and can be referred to for further details, time parallel methods can be classified into 4 different approaches, methods based on multiple shooting, domain decomposition and waveform relaxation, space-time multigrid and direct time parallel methods. Below a very brief overview of the main ideas behind these methods through some examples before taking a closer look at the strategy employed in this thesis.

Shooting type time parallel methods use a decomposition of the space-time domain into time slabs Ω_j, that is if $\Omega = \mathcal{S} \times [0, T]$ then $\Omega_j = \mathcal{S} \times [t_{j-1}, t_j]$ with $0 = t_0 < t_1 < .... < t_m = T$. Then there is usually an outer procedure that gives an approximated solution y_j for all $x \in \mathcal{S}$ at t_j, with $y_j = y(x, t_j)$, where y denotes solution (careful with approximation and solution...) which are then used to compute solutions in the time subdomains Ω_j independently and in parallel and give rise to an overall solution. One important example of how this can be done was given by Lions, Maday and Turinici in 2001, with an algorithm called parareal. A generalized version of it for a nonlinear problem of the form

$$y' = f(y), \quad y(t_0) = y_0$$

can be formulated as follows using two propagation operators:

1. $G(t_j, t_{j-1}, y_{j-1})$ is a coarse approximation of $y(t_j)$ with initial condition $y(t_{j-1}) = y_{j-1}$
2. $F(t_j, t_{j-1}, y_{j-1})$ is a more accurate approximation of $y(t_j)$ with the initial condition $y(t_{j-1}) = y_{j-1}$.

Starting with a coarse approximation Y_j^0 for all points in time t_j using G, the algorithm computes a correction iteration

$$Y_j^k = F(t_j, t_{j-1}, Y_{j-1}^{k-1}) + G(t_j, t_{j-1}, Y_{j-1}^k) - G(t_j, t_{j-1}, Y_{j-1}^{k-1})$$

Convergence? Extra work? General?

In **space-time domain decomposition methods** the idea is to divide the domain Ω into space slabs, that is $\Omega_i = \mathcal{S}_i \times [0, T]$ where $\mathcal{S} = \cup_{i=1}^n \mathcal{S}_i$. Then again an iteration or some other method is used to compute a solution on the local subdomains which can be done in parallel. A major challenge here is how to adequately deal with the values arising on the interfaces of the domain.

Direct Solvers in Space-Time. Here varying techniques are emplyed, one example is a method introduced in 2012 by S. Güttel called ParaExp, it is only applicable to linear initial value problems and most suitable for hyperbolic equations, where other time parallel solvers often have difficulties. To understand the underlying idea let us consider the following problem:

$$y'(t) = Ay(t) + g(t), \quad t \in [0, T], \quad u(0) = u_0 \tag{3.1}$$

One then considers an overlapping decomposition of the time interval $0 < T_1 < T_2 < .... < T_m = T$ into subintervals $[0, T_m], [T_1, T_m], [T_2, T_m], ..., [T_{m-1}, T_m]$. Now there are two steps to be performed. First solves a homogenous problem for the initial parts of each subdomain, that is $[0, T_1], [T_1, T_2], ..., [T_{m-1}, T_m]$, which is non-overlapping and can therefore be done in parallel:

$$v_j'(t) = Av_j(t) + g(t), \quad v_j(T_{j-1}) = 0 \quad t \in [T_{j-1}, T_j] \tag{3.2}$$

and afterwards the overlapping homogeneous problem is solved:

$$w_j'(t) = Aw_j(t), \quad w_j(T_{j-1}) = v_{j-1}(T_{j-1}), \quad t \in [T_{j-1}, T] \tag{3.3}$$

Due to linearity the overall solution can be obtained through summation

$$y(t) = v_k(t) + \sum_{j=1}^{k} w_j(t) \quad \text{with } k \text{ s.t. } t \in [T_{k-1}, T_k] \tag{3.4}$$

This way we obtain the general solution over the whole time interval. At first it is not clear why this approach gives a speed up since there is great redundency in the overlapping domains of the homogeneous problems which also need to be computed over big time intervals. The reason behind this is that the homogeneous problems can be computed very cheaply. They consist of matrix exponentials for which methods of near optimal approximations are known [[]].

In **space-time multigrid methods**, the parallelity comes from the discretisation of the space-time domain, that is considered as one, as mentioned before in the section on [FEM discretisation]. As a rather recent example of this type we will consider an approach by M. Gander and M. Neumüller [8]. Suppose we are considering a simple heat equation of the form $u_t - \Delta u = f$ and discretise it in a space - time setting using an implicit method like Backward Euler in time and another method, for example a discontinuous Galerkin approach in space. One then obtains a block triangular system of the following form

$$\begin{bmatrix} A_1 & & & & \\ B_2 & A_2 & & & \\ & B_3 & A_3 & & \\ & & ... & ... & \\ & & & B_m A_m & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ ... \\ u_m \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ ... \\ f_m \end{bmatrix} \tag{3.5}$$

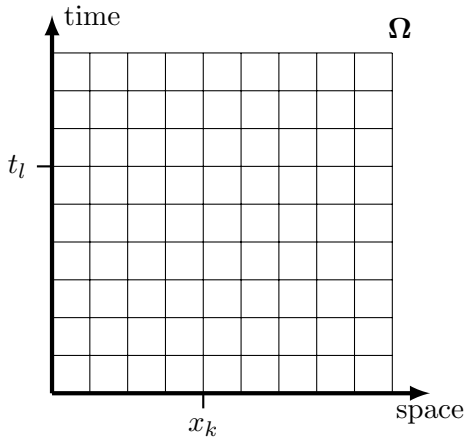
For the multigrid iteration they apply a block Jacobi smoother inverting each of the space blocks A_j before using a standard restriction operators in space - time to jump to a coarser grid, which is then being applied recursively. Choosing the right parameters one can achieve excellent scaling results for large systems.

Some solution approaches in space - time can be categorized in multiple approaches, for example is a 2 - level multigrid method starting with an initial guess obtained from the coarse grid and using an upwind (?) smoother the same as a simple parareal appraoch []

In this thesis we will consider a space-time multigrid approach but not of the previous type for the beforementioned symmetry reason (see section [...]) but instead go for a more straight forward space-time finite element assembly in addition to a first order least squares formulation that will be introduced in the following sections. For now we will only have a more detailed look at the space-time formulation that differs from a common finite element approach in the sense that our basis functions $\{\phi_1, ..., \phi_z\}$ which were introduced in section [...] are functions of time and space, i.e. $\phi_i = \phi_i(x, t)$. Hence it is possible to assemble one big system of equations that covers the entire space - time domain which can then be solved using a multigrid approach and differs from (2.15) in the sense that there are symmetric upper and lower off-diagonal blocks.

future influencing the past? upwind scheme?

The discretisation of the domain can be visualised as shown in figure [...].



Where one has $n + 1$ points in space and $m + 1$ points in time. The points (x_k, t_l) are then organized in the following manner, the i-th entry references $i = (n + 1) \cdot l + k$. That is we first run through all elements of a certain time step before moving on to the next time step which will then be assembled into one overall system. Details of how this is done will follow in the multigrid section. For simplicity we will only consider equidistant grids in each direction, where h_t denotes the size (better word?) of a mesh cell in time direction and h_{x_i} denote the size in the respective space dimension.

——— short transition ——— different ways to discretise the eqn on the mesh ——— one class of possibilities FEM ——

3.2 Newton's Method

3.2.1 Derivation

[source Briggs] One of the most well-known and most commonly used method to solve a non-linear problem is Newton's method which we will also be employed here and which we will derive in the following section and briefly discuss its advantages and disadvantages. So suppose we have an equation $F : \Omega \to \mathbb{R}^n$, with $F(x) = [0, 0, ..., 0]^T$, $x \in \mathbb{R}^n$, and $F \in C^1$, where we would like to determine the unknown root x. We consider a Taylor series expansion for an initial guess x:

$$F(x + h) = F(x) + \nabla F(x) \cdot h + o(||h||^2) \quad x \in \Omega. \tag{3.6}$$

If we now neglect the higher order terms, setting $F(x+h) = 0$ and replacing it by its first order Taylor approximation $F(x) + \nabla F(x) \cdot h$, which we can then solve for h, assuming that $\nabla F(x)$ is non-singular and use the result to update our initial guess x we obtain

$$x = x - [\nabla F(x)]^{-1} F(x). \tag{3.7}$$

(maybe better to write as an iteration)

3.2.2 Convergence

The convergence of Newton's method under certain conditions in the one dimensional case is established and proven relatively easily, however it requires much more work to establish the existence of and convergence to a root in the multidimensional case. There are different ways to do so, requiring (slightly) varying assumptions and hence obtaining (slighlty) different results. Here we will restrict ourselves to a classical version which was first shown by L. Kantorovich in 1940 and states the following:

Theorem 1. *Newton-Kantorovich.*
State here? Not too extensive?

While this theoretically establishes conditions for convergence, it is in practice ususally impossible to know if the necessary criteria are met. What we can however say, is that the likelyhood of converging at all and maybe even to the desired solution usually increases drastically if our initial guess is relatively close to the solution, assuming that "close" is defined in a meaningful way. Therefore one often solves a simplified related problem, commonly some sort of linearisation of the original problem which hopefully admits a similar solution that can then serve as a first initial guess. We will see in the following sections that the same idea will be applied here.

Or better something from Deuflhard here?!

LINE SEARCH, globalisation, hackbusch, trust region

— short transition section — So in each Newton step we consider a linearisation of our problem over the entire space - time domain, that we would like to be able to tackle at once and not solve for each time step consecutively. The following section gives an insight of how this can be done and explain why this might be a good idea —

3.3 Finite Element Methods

In order to find a numerical estimate to the solution of our partial differential equation we need a way to approximate the operators involved. And while there are many different ideas of how to do so the one we have chosen to use here is a finite element approach. They have shown to be a very powerful ... and are based on ... [source ?]
Gunzberger, LSFEM, preface : "since their emergence in the early 1950s, fem have become one of the most versatile and powerful methodologies for the approsimate numercial solutioooon of pdes."
"a fem is first and foremost a quasi-projection scheme. ... approximate solutions are then characterized as quasi-projections of the exact (weak) solutions onto the closed subspace.
FEM: do not directly deal with directly deal differential operators but through weak formulations leads to functional approximation as opposed to operator approximation

3.3.1 General Setting

The purpose of the subsequent section is not to establish the whole finite element framework from scratch but rather to provide the introduction of a unified notation that will be referred to throughout this thesis, a recollection of the most important properties needed. The foundation of every finite element formulation is finding an appropriate weak formulation which includes the choice of suitable trial and solution spaces. This is especially applicable in the case of a least squares approach and will be discussed in further detail in section [...].
Given Banach spaces X and Y, a bounded linear operator $\mathcal{A} : X \to Y$, $f \in Y$, we consider the problem:

$$\text{Find } u \in X \text{ such that} \quad \mathcal{A}u = f \text{ in } Y. \tag{3.8}$$

(when do we have existence and uniqueness of solutions ...?!) We are interested in the case where $\mathcal{A}$ represents a partial differential operator. As mentioned before the process of discretisation begins with turning (3.6) into a suitable variational equation which is defined in terms of two Hilbert spaces V and W, a continuous bilinear form $a(\cdot,\cdot) : V \times W \to \mathbb{R}$, and a bounded linear functional $L_f(\cdot) : W \to \mathbb{R}$ and is given by

$$\text{Find } u \text{ in } V \text{ such that: } a(u,v) = L_f(v) \quad \forall v \in V \tag{3.9}$$

An operator equation such as (3.6) may be reformulated into several different variational equations. We can see that we were originally seeking for a solution u in the space X whereas in the weak formulation one attempts to find a solution in the space V, that is generally not defined in X, and therefore often referred to as a weak solution. Hence the relationship between the spaces X, Y and V, W, and the operator $\mathcal{A}$ and the bilinear form $a(\cdot,\cdot)$ are of great importance, and while one generally wants the solution of the variational formulation (3.7) to be a "good" representation of the solution of the original problem (3.6), the definition of what that exactly means varies and usually depends on the nature of the problem and often some practicality issues. One possibility could be ... or too much? Therefore we have denoted them by the same letter but to be precise the solution u appearing in the subsequent paragraphs will always be referring to $u \in V$, because our aim now is to solve the variational formulation.
So let us assume for now that we have found a suitable weak formulation of the operator equation where trial and test space are equal, that is $V = W$. In addition to $a(\cdot,\cdot)$ being linear and bounded, which is equivalent to the continuity, we will also require it to be symmetric, hence we have more specifically that

$a(v_1, v_2) = a(v_2, v_1)$ for all $v_1, v_2 \in V$ (symmetry)
$a(v_1, v_2) \leq \beta ||v_1||_V \cdot ||v_2||_V$, for all $v_1, v_2 \in V$ and $\beta > 0$ (boundedness)
$a(v_1, v_1) \geq \alpha ||v_1||_V^2$, for all $v_1 \in V$ and $\alpha > 0$ (coercivity)

and $f \in V^*$, the dual space of V. Furthermore let us have homogeneous Dirichlet boundary conditions, that is $u = v = 0$ on $\partial\Omega$. Then by *Riesz representation theorem/Lax-Milgram* we obtain that there exists a unique solution $u \in V$ that solves (2.2). And additionally the existence of an operator $\tilde{\mathcal{A}} : V \to V^*$ given by

$$\mathcal{A}(u, v) = \langle \tilde{\mathcal{A}} u, v \rangle_{V^*, V} \quad \forall u, v \in V \tag{3.10}$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing (more...) between V and its dual space $V*$. Likewise we obtain for $L_f(\cdot)$ the existence of an unique (!) element $\tilde{f}$ through the relation

$$L_f(v) = \langle \tilde{f}, v \rangle_{V^*, V} \quad \forall v \in V \tag{3.11}$$

The variational formulation is therefore equivalent to the problem

$$\text{Find } u \in V \text{ such that} \quad \tilde{\mathcal{A}} u = \tilde{f} \quad \text{in } W^* \tag{3.12}$$

In the special case that $X = U$ and $Y = W^*$ we have that $\mathcal{A} = \tilde{\mathcal{A}}$ and $f = \tilde{f}$ but this is generally not the case.

3.3.2 Discretisation

A key element to actually finding a good approximation u^h of u is to choose a suitable finite dimensional (sub)space V_h where we search for the solution. We will consider a *Galerkin approach*, where we indeed have $V_h \subset V$, which itself is again a Hilbert space and therefore the projected finite dimensional problem called Galerkin equation looks as follows

$$\text{Find } u_h \text{ in } V_h \text{ such that: } a(u_h, v_h) = L_f(v_h) \quad \forall v_h \in V_h \tag{3.13}$$

and has a unique solution itself. Since (2.2) holds for all $v \in V$ it also holds for all $v \in V_h$, and hence $a(u - u_h, v_h) = 0$, a key property known as Galerkin orthogonality. With respect to the energy norm induced by $a(\cdot, \cdot)$, u_h is a best approximation to u, in the sense that

$$\begin{aligned} ||u - u_h||_a^2 &= a(u - u_h, u - u_h) = a(u - u_h, u) + a(u - u_h, v_h) \\ &\leq ||u - u_h||_a \cdot ||u - v_h||_a \quad \forall v_h \in V_h. \end{aligned} \tag{3.14}$$

We derive the third term from the second by using the Galerkin orthogonality. If we now divide both sides by $||u - u_h||_a$, we obtain that $||u - u_h||_a \leq ||u - v_h||_a$ for all $v_h \in V_h$. We also have an estimate on $u - u_h$ in terms of the norm $|| \cdot ||_V$. Using the coercivity constant α and the bound from above β, we see that

$$\begin{aligned} \alpha ||u - u_h||_V^2 &\leq a(u - u_h, u - u_h) = a(u - u_h, u - u_h) = a(u - u_h, u + v_h - v_h - u_h) \\ &= a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h) = a(u - u_h, u - v_h) \\ &\leq \beta ||u - u_h||_V \cdot ||u - v_h||_V \quad \forall v_h \in V_h. \end{aligned} \tag{3.15}$$

Dividing by $\alpha ||u - u_h||$ we have shown *Céa's lemma*, which states that (accuracy ... constant thing):

$$||u - u_h||_V \leq \inf_{v_h \in V_h} \frac{\beta}{\alpha} ||u - v_h||_V, \quad u \in V, u_h \in V_h \tag{3.16}$$

where u is the solution to (2.2) and u_h to the corresponding finite dimensional problem (2.3). Hence accuracy of our approximation depends in this case on the constants α and β.

If we assume that we have a discretisation Ω_h of our domain Ω, where $h > 0$ is a parameter depending on the mesh size. We furthemore want to assume that as h tends to zero this implies that $dim(V_h) \Rightarrow \infty$. Additionally let $\{V_h : h > 0\}$ denote a family of finite dimensional subspaces of V, for which we assume that

$$\forall v \in V : \quad \inf_{v_h \in V_h} ||v - v_h||_V \to 0 \text{ as } h \to 0. \tag{3.17}$$

That is with a mesh size tending to zero there exist increasingly precise approximations for every $v \in V$, whose infimum tends to zero as the mesh size does. But then we can also conclude by the beforementioned properties (3.24) and (3.25) that $||u - u_h||_V \to 0$ as $h \to 0$. Hence our approximate solution u_h will converge to the weak solution u.

3.3.3 Matrix Formulation

After establishing these theoretical properties our aim is now to construct a linear system of equations that can be solved efficiently. Since V_h is a finite dimensional Hilbertspace, it has a countable basis $\{\phi_1, \phi_2, ..., \phi_n\}$ and we can write every element in V_h as a linear combination of such, that is we also have $u_h = \sum_{j=1}^{n} u_j \phi_j$, where $u_1, ..., u_n$ are constant coefficients. Writing (3.21) in terms of the basis we obtain by linearity

$$a(\sum_{j=1}^{n} u_j \phi_j, \phi_i) = \sum_{j=1}^{n} u_j a(\phi_j, \phi_i) = L_f(\phi_i) \quad \forall \phi_i, \ i = 1, 2, ..., n \tag{3.18}$$

If we now write this as a system of the form $A_h U_h = L_h$ with entries entries $(A_h)_{ij} = a(\phi_j, \phi_i)$, $(L_h)_i = L_f(\phi_i)$, enough for basis Therefore each matrix entry represents the evaluation of an integral expression.

The question of how to choose favorable subspaces V_h, and a suitable basis for it has no trivial answer and depends on many factors and goes hand in hand with the question of how to best discretise the domain. Generally it seems like a sensible aim to opt for easily computable integrals giving rise to a linear system that is in turn as easy as possible to solve. Hence one objective might be to choose the basis $\{\phi_1, ..., \phi_n\}$ such that $supp(\phi_i) \cap supp(\phi_j) = \emptyset$ for as many pairs (i, j) as possible. Since this would ideally give rise to a sparse system of equations. It is also worth noting that due to the symmetry of $a(\cdot, \cdot)$, we have that $a_{ij} = a_{ji}$.

Already be more concrete here or wait until least squares section? — If one considers $V = H_0^1(\Omega)$, subspaces of polynomials, requirements on f ...

3.4 Least Squares Finite Element Methods

In this section which is based on ([9], mainly ch. 2.1) we would like to introduce least squares finite element methods (LSFEMs), a class of methods for finding the numerical solution of partial differential equations that incorporates two main ideas concepts; the concept of finite elements and optimisation problems. They are based on the minimisation of functionals which are constructed from residual equations. Historically finite element methods were first developed and analysed for problems like linear elasticity whose solutions describe minimisers for convex, quadratic functionals over infinite dimensional Hilbert spaces and therefore actually emerged in

an optimisation setting. A Rayleigh-Ritz approximation of solutions of such problems is then found by minimising the functional over a family of finite dimensional subspaces. For these classical problems the Rayleigh-Ritz setting gives rise to formulations that have a variety of favourable features and therefore have been and continue being highly successful. Among those are that:

1. *general regions* and boundary conditions can be treated in a systematic way relatively easily
2. conforming finite element spaces are sufficient to guarantee stability and optimal accuracy of the approximate solutions
3. all variables can be approximated using the same finite element space, e.g. the space of degree n piecewise polynomials on a particular grid
4. the arising linear systems are
   (i) sparse
   (ii) symmetric
   (iii) positive definite

Hence finite element methods originally emerged in the environment of an optimisation setting but have since then been extended to much broader classes of problems that are not necessarily associated to a minimisation problem anymore and generally lose the desirable features of the Rayleigh-Ritz setting except for 1 and 4 (iii). Least squares finite element methods can be seen as a new attempt to re-establishing as many advantageous aspects of the Rayleigh-Ritz setting as possible if not all for more general classes of problems. In the following section we will have a look at a classical straightforward Rayleigh-Ritz setting to familiarise ourselves with the set up before extending it to the more complicated class of problems introduced in [...].

We will consider a similar set up as in the finite element section (3.3.1) but with X and Y being Hilbert spaces, $f \in Y$ and a bounded, coercive linear operator $\mathcal{A} : X \to Y$, that is for some $\alpha, \beta > 0$:

$$\alpha||u||_X^2 \leq ||\mathcal{A}u||_Y^2 \leq \beta||u||_X^2 \quad \forall u \in Y \tag{3.19}$$

we consider the problem and the least squares functional:

$$\text{Find } u \in X \text{ such that } \mathcal{A}u = f \text{ in } Y \tag{3.20}$$

$$J(u; f) = ||\mathcal{A}u - f||_Y^2 \tag{3.21}$$

which poses the minimisation problem:

$$\min_{u \in X} J(u; f) \tag{3.22}$$

where we can see that the least squares functional (3.21) measures the residual of (3.20) in the norm of Y while seeking in for a solution in the space X. It follows that if a solution of the the problem (3.20) exists it will also be a solution of the minimisation problem. And a solution of the minimisation problem due to the definition of a norm will be a solution to (3.20) if the minimum is zero. If we consider $f = 0$, and using (3.19) we obtain that

$$\alpha^2||u||_X^2 \leq J(u; 0)||_Y^2 \leq \beta^2||v||_X^2 \quad \forall u \in X \tag{3.23}$$

a property of $J(\cdot,\cdot)$ which we will call norm equivalence, which is an important property when defining least squares functionals. We can derive a candidate for a variational formulation of the following form

$$a(u,v) = (\mathcal{A}u, \mathcal{A}v)_Y \text{ and } L_f(v) = (\mathcal{A}v, f)_Y \quad \forall u, v \in X \tag{3.24}$$

where $(\cdot,\cdot)_Y$ again denotes the innerproduct on Y, which will turn out to have all the desired properties. The operator form of (3.21) in the least squares setting is equivalent to the normal equations

$$\mathcal{A}^*\mathcal{A}u = \mathcal{A}^*f \quad \text{in } X \tag{3.25}$$

and corresponds to equation (3.9), with $\tilde{\mathcal{A}} = \mathcal{A}^*\mathcal{A}$, $\tilde{f} = \mathcal{A}^*f$ and $\mathcal{A}^*$ being the adjoint operator of $\mathcal{A}$. We can then move on to limiting our problem to a finite dimensional setting, where we choose a family of finite element subspaces $X^h \subset X$, parametrised by h tending to zero and restricting the minimisation problem to the subspaces. The LSFEM approximation $u^h \in X^h$ to the solution $x \in X$ of the infinite dimensional problem is the solution of the discrete minimisation problem

$$\min_{u^h \in X^h} J(u^h; f) \tag{3.26}$$

which is due to the fact that X^h is again a Hilbert space and therefore the same properties hold. Similarly to section (3.3.3) we can choose a basis $\{\phi_1, ..., \phi_n\}$ of X^h and will then obtain for the elements of $A^h\mathbb{R}^{n\times n}$, and $L_f^h \in \mathbb{R}^n$ that

$$A_{ij}^h = (\mathcal{A}\phi_j, \mathcal{A}\phi_i)_Y \quad \text{and} \quad (L_f^h)_i = (\mathcal{A}\phi_i, f)_Y \tag{3.27}$$

The following theorem establishes that this problem formulation actually gives rise to finite element set up.

Theorem 2. *Let $\alpha||u||_X^2 \leq ||\mathcal{A}u||_Y^2 \leq \beta||u||_X^2$ for all $u \in X$ hold, under the same assumptions as established in this section and let $X^h \subset X$. Then,*

(i) the bilinear form $a(\cdot,\cdot)$ defined in (3.21) is continuous, symmetric and coercive

(ii) the linear functional $L_f(\cdot)$ defined in (3.21) is continuous

(iii) the variational formulation (3.21) is of the form (3.9) and has a unique solution $u \in X$ which is also the unique solution of the minimisation problem (3.19)

(iv) there exists a constant $c > 0$, such that u and u_h satisfy

$$||u - u^h||_X \leq c \inf_{v^h \in X^h} ||u - v^h||_X \tag{3.28}$$

(v) the matrix A^h is symmetric positive definite

Idea of Proof: The properties *(i)* and *(ii)* directly follow from the boundedness and coercivity of $\mathcal{A}$ as well as the linearity of the inner product. Property *(iii)* follows from the theorem of Lax-Milgram while property *(iv)* is a consequence of Céa's lemma. The last property directly follows from the definition of A^h.

We therefore obtain that this least squares problem formulation has all the advantageous featurs of the Raleigh-Ritz setting without requiring $\mathcal{A}$ to be self-adjoint or symmetric which was our

initial goal. However it is worth noting that the differential operator $\tilde{\mathcal{A}} = \mathcal{A}^*\mathcal{A}$ is of higher order than the one in the original formulation, which therefore requires higher regularity assumptions which might be unpreferrable as well as impractical. Potential ways to overcome this problem will be discussed in the following section as it is also an issue that arises in the formulation of (....).
space-time LSFEM are not commonly used, future will tell if they establish themselves, what and how to compare to, what convergence to expect,

——- potentially short transition back to Newton —- Now this was considering a linear problem where f is independent of u but we will see later that these linear problems correspond to individual iterates of the Newton method.Hence if $f = f_u$ for a current iterate of u we know that each linearised problem has a unique solution. Therefore if the Newton iteration converges ... ?

Chapter 4

Problem Formulation

4.1 Set Up

In this chapter we would like to tie the beforementioned concepts together to derive a problem formulation that can subsequently be turned into an algorithm solving partial differential equations of the type introduced in the prologue and will be formulated more precisely. In order to do so we will firstly set the ground for the overall framework we are looking at. We consider a space-time domain

$$\Omega = \mathcal{S} \times \mathcal{T}, \quad \mathcal{T} = (0, T),\ T > 0 \text{ and } \mathcal{S} \subset \mathbb{R}^N \tag{4.1}$$

where $\mathcal{T}$ represents the time domain and $\mathcal{S}$ is the domain in space which we require to be Lipschitz regular. We may have a mixture of Dirichlet and Neumann boundary conditions on the boundary of Ω which we will denote by Γ and are labeled as $\Gamma_D, \Gamma_N \subset \Gamma$ respectively. We further assume them to be such that the problem is well posed. The class of partial differential equations introduced in the prologue that we would like to solve for then reads as the following:

$$\begin{aligned} u_t - \nabla \cdot (D(x)\nabla u) &= f(u) && (x,t) \in \Omega \\ u &= g_D && (x,t) \in \Gamma_D \\ \nabla u \cdot n &= g_N && (x,t) \in \Gamma_N \end{aligned} \tag{4.2}$$

It describes a parabolic partial differential equation with a non-linear right hand side. Typically we will have that $u(x, 0) = u_0 = g_D$ for all $x \in \mathcal{S}$ and Neumann boundary conditions on the boundary of $\mathcal{S}$ for $t \in (0, T)$.

The next step will be to derive an equivalent optimisation problem whose solution therefore then coincides with the solution of (4.2) at least in a weak sense (see for a further discussion of this ...). While we saw in the previous section that it is possible to derive a least squares formulation that recovers the properties of the Rayleigh- Ritz setting without requiring too many assumptions, this does not mean that the resulting formulation will necessarily also be practical, where practical means that the discretisation is relatively easy to implement, efficient and robust while still maintaining a sufficient level of accuracy. Therefore to make the methodology of LSFEMs competitive compared to other approaches like Galerkin approximations further considerations need to be taken into account. One hindrance one often encounters is the beforementioned higher order operator arising in (3.25), that would require a higher regularity on the function space X. When considering a simple Poisson equation with Dirichlet boundary conditions this would for example imply that we would require the solution u to be from H_0^2, instead of H_0^1 for which does not only limit the set of admissible solutions much more but for which is much harder to construct appropriate finite dimensional subspaces and therefore impractical to use.

In order to succomb this obstacle we will recast (4.2) as a system of coupled of equations only containing first order derivatives

$$\begin{aligned} u_t - div(\sigma) &= f(u) && (x,t) \in \Omega \\ \sigma &= D(x)\nabla u && (x,t) \in \Omega \\ u &= g_D && (x,t) \in \Gamma_D \\ \nabla u \cdot n &= g_N && (x,t) \in \Gamma_N \end{aligned} \tag{4.3}$$

Hence we can apply the methodologies introduced in section (3.4) at the price of introducing an additional variable. This way we can hopefully avoid having to use Sobolev spaces of order higher than one for our trial space, that is we can use $(\sigma, u) \in H^1_{div}(\Omega) \times H^1$. At the same time we would also like to stay away from Sobolev spaces of negative or fractional powers in the definition of the functional because they also complicate the computations. Nevertheless it is still not clear how indeed we can then choose the spaces appropriately. In order for theorem 2 (from the previous section, p. ...) to hold we require norm-equivalence, which can sometimes conflict with the aim for practicality and is a reoccuring problem in LSFEM. For further discussions we refer for example to chapter 2 of [9] and assume in this thesis that we have $f \in L^2(\Omega)$ which makes $Y = L^2(\Omega)$ a promising candidate for the definition of the least-squres functional. There are different ways to treat the boundary conditions in least-squares formulations. One possibility is to also include them in the functional as an additional term while another one would be to directly include them in the discretised system of the space. The former can lead to yet further complications when it comes to the computations because we also need to define an appropriate norm here as well as requiring the treatment of an additional term. Therefore we will assume here that the boundary is sufficiently regular and that the appropriate conditions can directly be imposed as part of the discretised system which will be discussed in more detail in the implementation section. Hence it now seems that we can define the functional J in the following way

$$\min_{\substack{\sigma \in H^1_{div}(\Omega) \\ u \in H^1(\Omega)}} J(\sigma, u) = \frac{1}{2} c_1 ||u_t - div(\sigma) - f(u)||_{L^2(\Omega)} + \frac{1}{2} c_2 ||\sigma - D(x) \cdot \nabla u||_{L^2(\Omega)} \tag{4.4}$$

We can see that we additionally have scaling parameters. The factors of one half are introduced so that the derivatives of J that we will be considering do not have an additional factor of two and the constants c_1 and c_2 can be used to give different weightings to the two terms without changing the actual minimiser at zero. This can ... It is also worth noting that in contrast to the example in section (3.4) $f = f(u)$ generally making the problem nonlinear. As mentioned in the prologue we consider an iterative approach to solve this problem and will therefore consider linearisations of the problem. But before going into more detail about that, let us consider the variational formulation of the coupled reaction-diffusion system (4.3)
The associated bilinear form that we obtain looks as follows (what about c_1, c_2?)

$$\mathcal{B}([\sigma, u], [\tau, v]) = \left(\begin{pmatrix} I & -D\nabla \\ -div & \frac{\partial}{\partial t} \end{pmatrix} \begin{pmatrix} \sigma \\ u \end{pmatrix}, \begin{pmatrix} I & -D\nabla \\ -div & \frac{\partial}{\partial t} \end{pmatrix} \begin{pmatrix} \tau \\ v \end{pmatrix} \right) \tag{4.5}$$

We can see that $\mathcal{B}$ is symmetric in the sense that $\mathcal{B}([\sigma, u], [\tau, v]) = \mathcal{B}([\tau, v], [\sigma, u])$. If we assume Dirichlet boundary conditions we also have that $\mathcal{B}$ is coercive.
TODO: discussion functions spaces
approximate by finite dimensional subspace

4.2 Least Squares in Space - Time

4.3 Linearisation

QUESTION OF WHERE TO LINEARISE ...
So then we do actually end up with a system where we have, for $x = (\sigma, u)$, $w = (\tau, v)$: $\mathcal{A}(x, w) = \mathcal{F}_k(w)$ variational equation in each step?!
As we have seen in the above considerations, we have that A is symmetric positive definite. And we can then consider

$$\min_{x \in X} J_k(x, F_k) = ||Ax - F_k||^2$$

4.4 Overall Problem

"In particular, for linear PDEs, residual minimization can lead to unconstrained optimization problems for convex quadratic functionals even if the original equations were not at all associated with optimiza- tion. If the PDE problem is nonlinear, then properly executed residual minimization leads to unconstrained minimization problems whose linearization22 gives rise to unconstrained minimization problems with convex quadratic functionals." (LSFEM book p.50) . Now we/I formulate as an optimisation problem where we would like to find the minimum over all admissible $u \in X$ where the function space X has to be defined appropriately such that the solution to the above equation corresponds to
Hence we have that a minimiser to the above formulation is at the same time a solution to our original problem. One can easily see that we have $J(\sigma, u) \geq 0$ for all $(\sigma, u) \in X_1 \times X_2$. Hence if $J(\sigma, u) = 0$ we must be at a minimum. The general strategy for solving these type of optimisation problems is more broad though, the idea is to find a pair $(\sigma, u) \in X_1 \times X_2$ for which $\nabla J(\sigma, u) = 0$ and $\nabla^2 J(\sigma, u)$ is positive definite which must consequently mean that (σ, u) is a minimiser.
Something like: We have seen above that every solution to the original problem is a solution to the minimisation problem and therefore if the solution to the original problem exists and is unique this one must be as well. Subsequently we proceed by determining the gradient and hessian of J and as we will see later are at the same time deriving a weak formulation that we will attempt to solve.
In the following I will denote $|| \cdot ||_{L^2(\Omega \times (0,T))}$ by $|| \cdot ||_2$ for brevity. choice of norms, why is this equivalent

4.5 Derivation of the Derivatives

In this section we will derive the first and second order directional partial derivatives of J with respect to its two variables in order to then set them to zero. The problem at hand is more complicated than in the standard case because we also have to take into account that the right hand side f is dependent on u and will therefore also appear in the differentiation. We can generally assume that the first and second order derivatives of J exist and are continuous almost everywhere since $|| \cdot ||_2$ is "continuouly differentiable" and $\sigma \in H^1_{div}(\Omega)$ and $u \in H^1(\Omega)$. Something no second order derivatives. In the following we will be determining the Gateaux derivatives of J, where we will be splitting the functional in three different terms that will be considered separately for greater clarity dividing it into the linear and nonlinear terms which is possible due to the linearity of the inner product.

The following notation will be used subsequently $||x||_2^2 = \langle x, x\rangle$ and $x = (\sigma, u)$, $h = (\tau, v)$, $k = (\rho, w)$. So let us consider the following

$$\begin{aligned} J_1(\sigma, u) &= \frac{1}{2} c_1 \langle u_t - div(\sigma), u_t - div(\sigma)\rangle \\ J_2(\sigma, u) &= \frac{1}{2} c_1 \langle 2u_t - 2div(\sigma) - f(u), -f(u)\rangle \\ J_3(\sigma, u) &= \frac{1}{2} c_2 \langle \sigma - \beta\nabla u, \sigma - \beta\nabla u\rangle \end{aligned} \tag{4.6}$$

where we can see that $J(\sigma, u) = J_1(\sigma, u) + J_2(\sigma, u) + J_3(\sigma, u)$. Now taking the partial directional derivatives we obtain, again taking the linearity of the inner product as well as its symmetry into account that

$$\begin{aligned} \frac{\partial J_1}{\partial \sigma} &= \lim_{\epsilon\to 0} \frac{J_1(\sigma + \epsilon\tau, u) - J_1(\sigma, u)}{\epsilon} \\ &= \lim_{\epsilon\to 0} \frac{c_1}{2\epsilon}(\langle u_t - div(\sigma + \epsilon\tau), u_t - div(\sigma + \epsilon\tau)\rangle - \langle u_t - div(\sigma), u_t - div(\sigma)\rangle) \\ &= \lim_{\epsilon\to 0} \frac{c_1}{2\epsilon}(\langle u_t, u_t\rangle - \langle u_t, div(\sigma)\rangle - \epsilon\langle u_t, div(\tau)\rangle - \langle div(\sigma), u_t\rangle + \langle div(\sigma), div(\sigma)\rangle \\ &+ \epsilon\langle div(\sigma), div(\tau)\rangle - \epsilon\langle div(\tau), u_t\rangle + \epsilon\langle div(\tau), div(\sigma)\rangle + \epsilon^2\langle div(\tau), div(\tau)\rangle \\ &- \langle u_t, u_t\rangle + \langle u_t, div(\sigma)\rangle + \langle div(\sigma), u_t\rangle - \langle div(\sigma), div(\sigma)\rangle) \\ &= \lim_{\epsilon\to 0} \frac{c_1}{2\epsilon}(-2\epsilon\langle u_t, div(\tau)\rangle + 2\epsilon\langle div(\sigma), div(\tau)\rangle + \epsilon^2\langle div(\tau), div(\tau)\rangle) \\ &= -c_1\langle u_t, div(\tau)\rangle + c_1\langle div(\sigma), div(\tau)\rangle \end{aligned} \tag{4.7}$$

We can see that the terms only containing σ or u cancel. We end up with a number of mixed terms as well as the terms containing purely τ and v. Due to the factor of $\frac{1}{2}$ in front of the inner products in J and the symmetry of the inner product, the mixed terms add up 1 or -1 respectively. Again because of the bilinearity of the inner product we can write ϵ in front of the individual terms, often they will cancel with the factor of $\frac{1}{\epsilon}$ in front. If we now take the limit with respect to ϵ going to zero all terms with an ϵ in both arguments will tend to zero which gives us the remaining result. By proceeding analogously for equation J_2 and J_3 we obtain in these cases:

$$\begin{aligned} \frac{\partial J_2}{\partial \sigma} &= c_1\langle div(\tau), f(u)\rangle \\ \frac{\partial J_3}{\partial \sigma} &= c_2\langle \sigma, \tau\rangle - c_2\beta\langle \tau, \nabla u\rangle \end{aligned} \tag{4.8}$$

Let us now turn to the partial derivatives with respect to u. Here we obtain the following for J_1 and J_3:

$$\begin{aligned} \frac{\partial J_1}{\partial u} &= \lim_{\epsilon\to 0} \frac{J_1(\sigma, u + \epsilon v) - J_1(\sigma, u)}{\epsilon} \\ &= \lim_{\epsilon\to 0} \frac{c_1}{2\epsilon}(\langle (u + \epsilon v)_t - div(\sigma), (u + \epsilon v)_t - div(\sigma)\rangle - \langle u_t - div(\sigma), u_t - div(\sigma)\rangle) \\ &= c_1\langle u_t, v_t\rangle - c_1\langle v_t, div(\sigma)\rangle \end{aligned} \tag{4.9}$$

$$\frac{\partial J_3}{\partial u} = -c_2\beta\langle \sigma, \nabla v\rangle + c_2\beta^2\langle \nabla u, \nabla v\rangle$$

In the case of J_2, we have to take the non-linearity of f into account. If we assume that f sufficiently smooth (what do we need exactly?!) that is $\lim_{\epsilon\to 0} f(u+\epsilon v) = f(u)$ and $\lim_{\epsilon\to 0}\langle f(u+\epsilon v), f(u+\epsilon v)\rangle - \langle f(u), f(u)\rangle = \langle f'(u)\cdot v, f(u)\rangle + \langle f(u), f'(u)\cdot v\rangle$ which can be added due to symmetry.

$$\begin{aligned}
\frac{\partial J_2}{\partial u} &= \lim_{\epsilon\to 0}\frac{J_2(\sigma, u+\epsilon v) - J_2(\sigma, u)}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{c_1}{2\epsilon}(\langle 2(u+\epsilon v)_t - 2div(\sigma) - f(u+\epsilon v), -f(u+\epsilon v)\rangle - \langle 2u_t - 2div(\sigma) - f(u), -f(u)\rangle) \\
&= \lim_{\epsilon\to 0}\frac{c_1}{2\epsilon}(-2\langle u_t, f(u+\epsilon v)\rangle + 2\langle u_t, f(u)\rangle \\
&- 2\epsilon\langle v_t, f(u+\epsilon v)\rangle \\
&+ 2\langle div(\sigma), f(u+\epsilon v\rangle - 2\langle div(\sigma), f(u)\rangle \\
&+ \langle f(u+\epsilon v), f(u+\epsilon v)\rangle - \langle f(u), f(u)\rangle) \\
&= -c_1\langle u_t, f'(u)\cdot v\rangle - c_1\langle v_t, f(u)\rangle + c_1\langle div(\sigma), f'(u)\cdot v\rangle + c_1\langle f(u), f'(u)\cdot v\rangle
\end{aligned} \tag{4.10}$$

Hence we obtain the following partial first order directional derivatives.

$$J_\sigma[\tau] = \frac{\partial}{\partial\sigma}J(\sigma,u)[\tau] = c_2\langle\sigma,\tau\rangle + c_1\langle div(\sigma), div(\tau)\rangle - c_2\beta\langle\nabla u,\tau\rangle - c_1\langle u_t, div(\tau)\rangle - c_1\langle f(u), div(\tau)\rangle \tag{4.11}$$

$$\begin{aligned}
J_u[v] = \frac{\partial}{\partial u}J(\sigma,u)[v] =& c_1\langle u_t, v_t\rangle - c_1\langle v_t, div(\sigma)\rangle - c_2\beta\langle\sigma,\nabla v\rangle + c_2\langle\nabla u,\nabla v\rangle \\
&- c_1\langle u_t, f'(u)\cdot v\rangle - c_1\langle v_t, f(u)\rangle - c_1\langle div(\sigma), f'(u)\cdot v\rangle + c_1\langle f(u), f'(u)\cdot v\rangle
\end{aligned} \tag{4.12}$$

Following the same principles one can determine the second order partial derivatives whose derivation will only be briefly outlined here for the most difficult terms which are those including f.

$$\frac{\partial^2}{\partial\sigma^2}J[\tau][\rho] = c_2\langle\rho,\tau\rangle + c_1\langle div(\rho), div(\tau)\rangle \tag{4.13}$$

$$\frac{\partial^2}{\partial\sigma\partial u}[v][\tau] = \frac{\partial^2}{\partial u\partial\sigma}[\tau][v] = -\langle\tau,\nabla v\rangle - \langle v_t, div(\tau)\rangle - \langle div(\tau), f'(u)v\rangle \tag{4.14}$$

$$
\begin{aligned}
\frac{\partial^2 J}{\partial u^2}[v][w] &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}(J_u(\sigma,u+\epsilon w)[v]-J_u(\sigma,u)[v])\\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}(c_1\langle(u+\epsilon w)_t,v_t\rangle + c_2\langle\nabla(u+\epsilon w),\nabla v\rangle - c_1\langle(u+\epsilon w)_t,f'(u+\epsilon w)\cdot v\rangle\\
&- c_1\langle v_t,f(u+\epsilon w)\rangle - c_1\langle div(\sigma),f'(u+\epsilon w)\cdot v\rangle + c_1\langle f(u+\epsilon w),f'(u+\epsilon w)\cdot v\rangle\\
&- J_u(\sigma,u)[v])\\
&= c_1\langle w_t,v_t\rangle + c_2\langle\nabla w,\nabla v\rangle\\
&+ \lim_{\epsilon\to 0}\frac{1}{\epsilon}(c_1\langle u_t,f'(u+\epsilon w)\cdot v\rangle - c_1\langle u_t,f'(u)\cdot v\rangle\\
&- \epsilon\cdot c_1\langle w_t,f'(u+\epsilon w)\cdot v\rangle\\
&- c_1\langle v_t,f(u+\epsilon w)\rangle + c_1\langle v_t,f(u)\rangle\\
&- c_1\langle div(\sigma),f'(u+\epsilon w)\cdot v\rangle + c_1\langle div(\sigma),f'(u)\cdot v\rangle\\
&+ c_1\langle f(u+\epsilon w),f'(u+\epsilon w)\cdot v\rangle - c_1\langle f(u),f'(u)\cdot v\rangle)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 J}{\partial u^2}[v][w] &= c_1\langle w_t,v_t\rangle + c_2\langle\nabla w,\nabla v\rangle + c_1\langle u_t,w^T f''(u)v\rangle - c_1\langle w_t,f'(u)\cdot v\rangle - c_1\langle v_t,f'(u)\cdot w\rangle\\
&- c_1\langle div(\sigma),w^T f''(u)v\rangle + c_1\langle f(u),w^T f''(u)v\rangle + \langle f'(u)\cdot w,f'(u)\cdot v\rangle
\end{aligned}
\tag{4.15}
$$

Talk about "derivatives in direction"
Then the whole Gateaux derivative thing and I still need a good explanation how the directional derivatives which in this case are functions become the basis or test functions.

We can reorganise the terms using the linearity of the differential operators as well as of the inner product.
Using the continuity of the .... If we also assume that f is Gateaux diff / cts / ... on measurable sets ... more specifics ... Only considering the terms containing f we can see that ...
In order to obtain the variational formulation of our problem
Least Squares space time, end up with a large system, makes sense to allow for parallelisation (?)

what is X?

show somehow that Thm 2.5 holds.

In my case ....

4.6 Overall Assembly

Chapter 5

Multigrid Methods

5.1 General Idea

"lose notation again, not associated anymore with .... "
Multigrid Methods is an important class of algorithms to iteratively solve/approximate linear systems of equations of the form $Au = f$ where $A \in \mathbb{R}^n$ is a (sparse) symmetric positive definite matrix.

It incorporates a variety of ideas that make use of Core concept :
Due to the structure of A its eigenvectors form an orthonormal basis of $\mathbb{R}^n$.
which allows us to consider the solution u, the approximate solution w as well as the difference between the two, the error $e = u - w$ as a linear combination of this basis. Furthermore ... frequency ... eigenfunctions ... eigenvectors. Therefore it is possible to differentiate between high and low frequency error where one generally assumes ... (further discussion of what high and low frequency is later ... physical connection to solution, AMG $\iff$ GMG)
Standard iterative solvers like Jacobi, Gauss-Seidel, ... (need reference here) are generally known to reduce the high frequency error quite well. (Intuitively) this makes sense since the operator the matrix A stems from/represents/is consists of local connections. sparsity of it. Local communication
$\Rightarrow$ property of A not exactly of smoother ?
Hence this type of solvers are usually refered to as smoothers or relaxation schemes. So if we therefore assume that they reduce the high frequency error quite efficiently it is seems natural to look for a way to effectively minimise the low frequency error.
some pictures?

If we suppose we were to project the low frequency error onto a coarse grid it would suddenly be of higher frequency compared to the grid. Obtain correction
nested meshes
elliptic problems
In geometric multigrid we have a predefined sequence of nested meshes of specific coarsening factors (that can vary in different directions (and on different levels)). The coarse level spaces still represent the original problem just with a lower resolution.
Algebraic multigrid on the other hand does not have predefined coarse level spaces but instead they are chosen according to a given rule, that takes the values of A (or other known properties of the problem) into account. This is favorable if .... but also more expensive to compute.

5.2 Basic Algorithm

As mentioned before there is unique way to construct the ideal operators necessary.
Below we can see a multigrid V-cycle iteration scheme. Where we assume that $J, J-1, ..., 0$ denotes the grid levels from finest to coarsest.

Multigrid V-cycle

Let w^J be the initial guess (on the finest grid level). Then repeat the following until convergence criterium is met or number of iterations exceeds a certain threshold:

- do ν_{J_a} smoothing steps on $A^J u^J = f^J$ with initial guess w^J
- compute $f^{J-1} = I_J^{J-1} r^J$
- do ν_{J-1_a} smoothing steps on $A^{J-1} u^{J-1} = f^{J-1}$ with initial guess $w^{J-1} = 0$ (vector)
- compute $f^{J-2} = I_{J-1}^{J-2} r^{J-1}$
- do ν_{J-2_a} smoothing steps on $A^{J-2} u^{J-2} = f^{J-2}$ with initial guess $w^{J-2} = 0$ (vector)
- compute $f^{J-3} = I_{J-2}^{J-3} r^{J-2}$

  ...
  ...
- solve $A^0 u^0 = f^0$

  ...
  ...
- correct $w^{J-2} = w^{J-2} + I_{J-3}^{J-2} w^{J-3}$
- do ν_{J-2_b} smoothing steps on $A^{J-2} u^{J-2} = f^{J-2}$ with initial guess w^{J-2}
- correct $w^{J-1} = w^{J-1} + I_{J-2}^{J-1} w^{J-2}$
- do ν_{J-1_b} smoothing steps on $A^{J-1} u^{J-1} = f^{J-1}$ with initial guess w^{J-1}
- correct $w^J = w^J + I_{J-1}^J w^{J-1}$
- do ν_{J_b} smoothing steps on $A^J u^J = f^J$ with initial guess w^J

picture V-cycle?

5.3 Convergence Properties and Complexity

strengthened Cauchy - Schwarz necessary?

Theorem 3. *Convergence.*

5.4 Algebraic Multigrid

5.4.1 Essential Concepts

Definition 1. *Strong dependence.*

As briefly mentioned before
Concept of strong dependence. Different ways to define this, most commonly ...?
This is where later on the adaption to the monodomain equation is made because here we can take specific knowledge about the equation into consideration.

5.4.2 Coarse Space Construction, Eigenvectors, ...

5.5 Non-Linear Multigrid

Chapter 6

Implementation and Numerical Results

Whole one dimensional thing therefore H^1 and H^{div} the same and hence $\nabla(u) = div(u) = \frac{\partial u}{\partial x}$ s
The Hessian that arises from the minimisation problem is not simply the discretisation of one differential operator as one would for example obtain when solving a Poisson equation. Instead it is a combination of the Laplacian operator in space, the first order advection in time and the linearised (and adapted) first and second order derivatives of $f(u)$.
Therefore it is complicated/not really possibly to relate to physical things .... ?
What is the idea?

The Newton iteration at hand looks as follows:

$$y_{k+1} = y_k - H_J^{-1}(y_k)(\nabla J(y_k)) \tag{6.1}$$

where $y_k = (\sigma_k, u_k)^T$ and the hessian H_J is defined as $H_J(y_k) = D^2 J(\sigma_k, u_k)$.
Written in matrix notation one obtains the following system which is split up in the linear and non-linear part:

$$\begin{aligned} \nabla J(y_k) = &\begin{bmatrix} -\langle u_t, div(\tau)\rangle + \langle \sigma_k, \tau\rangle + \langle div(\sigma_k), div(\tau) - \langle \nabla u_k, \tau\rangle \\ \langle u_t, v_t\rangle - \langle v_t, div(\sigma_k) + \langle \nabla u, \nabla v\rangle - \langle \sigma_k, \nabla v\rangle \end{bmatrix} \\ + &\begin{bmatrix} +\langle f(u_k), div(\tau)\rangle \\ \langle div(\sigma_k), f'(u_k)v\rangle + \langle f(u_k), f'(u_k)v\rangle - \langle (u_k)_t, f'(u_k)v\rangle - \langle v_t, f(u)\rangle \end{bmatrix} \end{aligned} \tag{6.2}$$

$$\begin{aligned} H_J(y_k) = &\begin{bmatrix} \langle \tau, \rho\rangle + \langle div(\tau), div(\rho)\rangle & -\langle \rho, \nabla v\rangle - \langle v_t, div(\rho)\rangle \\ -\langle \tau, \nabla w\rangle - \langle w_t, div(\tau) & \langle v_t, v_t\rangle + \langle \nabla v, \nabla w\rangle \end{bmatrix} \\ + &\begin{bmatrix} 0 & -\langle div(\tau), f'(u_k)\cdot w\rangle \\ -\langle div(\rho), f'(u_k)v\rangle & -\langle (u_k)_t, w^T f''(u_k)v\rangle + \langle div(\sigma_k), w^T f''(u_k)v\rangle + \langle f'(u_k)w, f'(u_k)w\rangle + ... \\ & ... + \langle f(u_k), w^T f''(u_k)v\rangle - \langle w_t, f'(u_k)v\rangle - \langle v_t, f'(u_k)w\rangle \end{bmatrix} \end{aligned} \tag{6.3}$$

Using basis and test functions of the type .... P1. Approximate the integrals using Gaussian quadrature of degree three, therfore computing integrals of polynomials of degree one, which includes our basis functions, however not exact for f which is represented as ... The system one obtains from the first
u as a sum of basis functions, σ as well

reorganise these terms to set up the Newton iteration how does my newton iteration look like exactly?
write that up ....
put in additional terms for equation, for now u_p is u from previous iteration. 3rd line will go on rhs, last line as well, 2nd line on the left

$$\begin{bmatrix} A_{\sigma\sigma} & \\ & A_{uu} \end{bmatrix} \cdot \begin{bmatrix} \sigma \\ u \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ diag(M_t \times G_h \cdot (f'(u) \cdot v)) & -diag(G_t \times M_h) \cdot \end{bmatrix} = \begin{bmatrix} -\langle div(\tau), f(u_p) \rangle \\ \langle v_t, f(u_p) \rangle \end{bmatrix}$$

Then leave it at the continuous problem formulation though.
Next thing that we need is the space-time discretisation.
what changes between Dirichlet and Neumann boundary conditions?
Boundary conditions are directly enforced. How to get symmetry back?
Some sort of convergence test?
Eventually come to multigrid implementation.
The chosen basis function are from Q_1, that is the space of For each basis function we have that and

$$\phi_{ij}(x_k, t_l) = \begin{cases} 1 & \text{if } (k,l) = (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

6.1 Newton Implementation

how to make this better ....? line search requires gradient evaluations in inner loop

Chapter 7

Conclusions and Outlook

Chapter 8

Epilogue

Bibliography

[1] "World healt organisation, cardiovascular diseases," Oct. 2018.

[2] W. Harvey, *Exerciatio anatomica de motu cordis et sanguinis in animalibus.* Frankfurt, 1628.

[3] P. C. Franzone, L. F. Pavarino, and S. Scacchi, *Mathematical cardiac electrophysiology*, vol. 13. Springer, 2014.

[4] A. C. of Cardiology, "cardiosmart, how the heart works," Oct. 2018.

[5] M. Potse, B. Dubé, J. Richer, A. Vinet, and R. M. Gulrajani, "A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2425–2435, 2006.

[6] P. Colli Franzone and L. F. Pavarino, "A parallel solver for reaction–diffusion systems in computational electrocardiology," *Mathematical models and methods in applied sciences*, vol. 14, no. 06, pp. 883–911, 2004.

[7] M. J. Gander, "50 years of time parallel time integration," in *Multiple Shooting and Time Domain Decomposition Methods*, pp. 69–113, Springer, 2015.

[8] M. J. Gander and M. Neumuller, "Analysis of a new space-time parallel multigrid algorithm for parabolic problems," *SIAM Journal on Scientific Computing*, vol. 38, no. 4, pp. A2173–A2208, 2016.

[9] P. B. Bochev and M. D. Gunzburger, *Least-squares finite element methods*, vol. 166. Springer Science & Business Media, 2009.