

formulation, that is a problem of the type (...), into a symmetric variational formulation, that is a symmetric bilinear form $a(\cdot, \cdot)$, which can subsequently be restricted to finite-dimensional subspaces and where we search for approximate solutions. However one possibility is through the differentiation of certain energy functionals, because we know by the theorem of Schwarz that order of differentiation with respect to partial derivatives is interchangeable and therefore leads to symmetry. How to construct these functional to be related to particular differential equations will be discussed in the following section.

3.2 Least Squares Finite Element Methods

In this section which is based on ([7], 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 $\mathcal{A} : X \rightarrow Y$, that is for some $\alpha, \beta > 0$:

$$\alpha \|u\|_X^2 \leq \|\mathcal{A}u\|_Y^2 \leq \beta \|u\|_X^2 \quad \forall u \in X \quad (3.12)$$

we consider the problem and the least squares functional:

$$\text{Find } u \in X \text{ such that } \mathcal{A}u = f \text{ in } Y \quad (3.13)$$

$$J(u; f) = \|\mathcal{A}u - f\|_Y^2 \quad (3.14)$$

which poses the minimisation problem:

$$\min_{u \in X} J(u; f) \quad (3.15)$$

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

$$\alpha^2 \|u\|_X^2 \leq J(u; 0) \leq \beta^2 \|v\|_X^2 \quad \forall u \in X \quad (3.16)$$

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

$$a(u, v) = (\mathcal{A}u, \mathcal{A}v)_Y \text{ and } L_f(v) = (\mathcal{A}v, f)_Y \quad \forall u, v \in X \quad (3.17)$$

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

$$\mathcal{A}^* \mathcal{A}u = \mathcal{A}^* f \quad \text{in } X \quad (3.18)$$

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

$$\min_{u^h \in X^h} J(u^h; f) \quad (3.19)$$

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 $\{\phi_1, \dots, \phi_n\}$ 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

$$A_{ij}^h = (\mathcal{A}\phi_j, \mathcal{A}\phi_i)_Y \quad \text{and} \quad (L_f^h)_i = (\mathcal{A}\phi_i, f)_Y \quad (3.20)$$

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

Theorem 1. *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

$$\|u - u^h\|_X \leq c \inf_{v^h \in X^h} \|u - v^h\|_X \quad (3.21)$$

(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 features 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 unpreferable 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 problem formulation of the subsequent chapter.

3.3 Space-Time Solution Methods

Most solution methods for partial differential equations do not use the time direction 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 [5]. However evolution over time behaves differently from the spatial dimensions, in the sense that it follows the causality principle. It means 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 [5], 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 Ω into time slabs Ω_j , that is if $\Omega = \mathcal{S} \times [0, T]$ where \mathcal{S} describes the spatial domain then $\Omega_j = \mathcal{S} \times [t_{j-1}, t_j]$ with $0 = t_0 < t_1 < \dots < t_m = T$. Then there is usually an outer procedure that gives a coarse approximated solution y_j for all $x \in \mathcal{S}$ at t_j for all j , 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,

Chapter 4

Derivation of a Least Squares Finite Element Space - Time Discretisation

4.1 Construction of an Equivalent Minimisation Problem

In this chapter we will tie the beforementioned concepts together to derive a discretised problem formulation that can subsequently be turned into an algorithm capable of approximating reaction diffusion equations. In order to do so let us firstly set the ground for the overall framework we are looking at. We consider a space-time domain

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

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. *Is this enough?* The class of partial differential equations introduced in the prologue that we would like to solve for then reads as the following:

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

It describes a parabolic partial differential equation with a non-linear right hand side where the divergence is defined as $\operatorname{div}(\sigma) = \frac{\partial \sigma}{\partial x_1} + \dots + \frac{\partial \sigma}{\partial x_N}$ for $\sigma = \sigma(x, t)$ and $\nabla u = [\frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_N}]^T$ for $u = u(x, t)$. And where we further assume that $D(x)$ is a bounded, symmetric positive definite matrix of size $N \times N$ with functions in $L^2(\mathcal{S})$ for almost all $x \in \mathcal{S}$. *Are these sufficient assumptions on $D(x)$?* 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. Since we are entirely working with finding solutions in Sobolev spaces in the least squares setting we can generally only require equivalence to a primal weak formulation of (4.3) and (4.2) or equivalence with respect to the solution space of the variational formulation, for a further discussion we refer to [source]. Generally when working with least squares finite element formulations choosing a suitable solution space V and data space Z is often non trivial as there are a number of difficulties that can arise. One usually faces a trade off between constructing a mathematically well-defined problem and

allowing for a relatively simple, efficient, robust while still accurate implementation. Therefore to make the methodology of LSFEMs competitive compared to other approaches like Galerkin approximations further considerations need to be taken into account. We saw in the previous chapter that it is possible to derive least squares formulations that recover the properties of the Rayleigh- Ritz setting, however one hindrance one encounters in this setting as well as in many others is the higher order operator arising in (3.25), that would require a solution space of higher regularity. 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 [7]. This does not only heavily limit the set of admissible solutions but it is additionally much harder to construct appropriate finite dimensional subspaces for and is therefore impractical to use. In order to succumb this obstacle we will recast (4.2) as a system of coupled equations only containing first order derivatives to apply the methodologies introduced in section (3.4) at the price of introducing an additional variable.

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

Rearranging this equation into a vector form we obtain in Ω

$$\begin{pmatrix} I & -D(x) \nabla \\ -\operatorname{div} & \frac{\partial}{\partial t} \end{pmatrix} \begin{pmatrix} \sigma \\ u \end{pmatrix} = \begin{pmatrix} 0 \\ f(u) \end{pmatrix} \tag{4.4}$$

which can be shortened to $\mathcal{A}([\sigma, u]) = \tilde{f}(u)$, where \mathcal{A} denotes the differential operator and $\tilde{f}(u)$ the right hand side.

Show that this operator is linear and bounded? but then norm ... spaces ... bla bla bla ...

Remark. It is not clear yet though in which space we will be looking for a solution. In general when working with strong and weak formulations there is a number of function spaces involved, problem formulations that are similar but not the same and it is difficult to represent them in clear but short notation. This can easily lead to some confusion. In the current case the situation is even more complicated since we are trying to derive a weak formulation in a least squares setting, while additionally there obviously exist the regular or so-called primal weak formulations of (4.2) which we will sometimes refer to. Therefore we will first derive a least functional J with a corresponding variational formulation, postponing the definition of appropriate spaces for now but simply assuming them to already be well-defined.

So instead of looking for a solution of the strong formulation, let us now turn to the derivation of an optimisation problem that we would like to satisfy in a weak sense. The properties that we would like to be fulfilled are the following. We want a solution of (4.3) to be a global minimum of the optimisation problem, independently of the choice of the spaces that we are using and its associated norm, hence we additionally want the least squares functional J to be zero for a solution of (4.3). On the other hand if $J(\sigma, u) = 0$ then the original problem (4.3) also has to be satisfied if only in a weak sense with respect to the spaces U and Z . We can check that all three properties hold for the functional

$$\tilde{J}(\sigma, u) = \|u_t - \operatorname{div}(\sigma) - f(u)\|_Z^2 + \|\sigma - D(x) \nabla u\|_Z^2 \tag{4.5}$$

Furthermore it has shown to be practical to be able to weight the two terms with coefficients [source] which does not affect the overall solution but grants us the possibility to numerically give more importance to one term than the other. We also introduce additional scalars of 0.5 to

simplify further computations of the derivatives as we will see in the following section. Therefore the the minimisation problem then reads

$$\min_{(\sigma, u) \in U} J(\sigma, u) = \frac{1}{2} c_1 \|u_t - \operatorname{div}(\sigma) - f(u)\|_Z^2 + \frac{1}{2} c_2 \|\sigma - D(x) \nabla u\|_Z^2. \quad (4.6)$$

J now defines an energy we can minimise over. If we consider the functional for $f = 0$ we obtain

$$J([\sigma, u], 0) = \frac{1}{2} c_1 \langle u_t - \operatorname{div}(\sigma), u_t - \operatorname{div}(\sigma) \rangle_Z + \frac{1}{2} c_2 \langle \sigma - D(x) \nabla u, \sigma - D(x) \nabla u \rangle_Z. \quad (4.7)$$

which more specifically gives rise to the following bilinear form if we differentiate J with respect to directional derivatives τ and v , and subsequently sum over them [see section 4.5 for the derivation].

$$\mathcal{B}([\sigma, u], [\tau, v]) = \left\langle \begin{pmatrix} I & -D\nabla \\ -\operatorname{div} & \frac{\partial}{\partial t} \end{pmatrix} \begin{pmatrix} \sigma \\ u \end{pmatrix}, \begin{pmatrix} I & -D\nabla \\ -\operatorname{div} & \frac{\partial}{\partial t} \end{pmatrix} \begin{pmatrix} \tau \\ v \end{pmatrix} \right\rangle_Z \quad (4.8)$$

The resulting candidate for a variational formulation can be derived as usual [source] that is as the *sum* of the directional derivatives of J , where the *directions* become the testfunctions.

$$\begin{aligned} &\text{Find } (\sigma, u) \in U \text{ such that } \mathcal{B}([\sigma, u], [\tau, v]) = L_{\tilde{f}}(u)(v) \quad \forall (\tau, v) \in W \\ &\text{with } \mathcal{B}([\sigma, u], [\tau, v]) = (\mathcal{A}[\sigma, u]^T, \mathcal{A}[\tau, v]^T) \text{ and } \mathcal{L}_{\tilde{f}}(u)([\tau, v]) = (\mathcal{A}[\tau, v], \tilde{f}(u))_Z \end{aligned} \quad (4.9)$$

Remark: $L_{\tilde{f}}(u)(v)$ is a linear operator in v but non-linear in u . What remains to be determined are the function spaces V , W , Z which potentially even lead us to a setting where can show norm equivalence in order for *Theorem 3* to hold.

4.2 Function Spaces

In order to define V and Z let us consider the optimisation problem [...] and its variational formulation [...] again. We would like to allow for the broadest class of solutions possible while still ensuring that all terms are well-defined in V , and staying away from Sobolev spaces of negative or fractional powers due to the beforementioned practicality reasons. Additionally to guarantee the existence of all terms involved we have to make sure that the present weak derivatives of σ and u exist in the induced inner product of Z . Let us therefore consider the following spaces

$$H_{div}^1(\mathcal{S}) = \{\sigma \in (L^2(\mathcal{S}))^n : \operatorname{div}(\sigma) \in L^2(\mathcal{S})\} \quad (4.10)$$

$$H_{DIV}^1(\Omega) = H_{div}^1(\mathcal{S}) \times L_2(\mathcal{T}) \quad (4.11)$$

where and when to use \times , overall notation

$$U = W = H_{DIV}^1(\Omega) \times H^1(\Omega) \quad (4.12)$$

$$Z = L^2(\Omega) \quad (4.13)$$

that is we require that

$$\sigma \in H_{DIV}^1(\Omega) \text{ and } u \in H^1(\Omega) \text{ with} \quad (4.14)$$

$$\|\sigma\|_{H_{DIV}^1(\Omega)} = \|\sigma\|_{L^2(\Omega)} + \|\operatorname{div}(\sigma)\|_{L^2(\Omega)}, \quad (4.15)$$

$$\|u\|_{H^1(\Omega)} = \|u\|_{L^2(\Omega)} + \|\nabla u\|_{L^2(\Omega)} + \|u_t\|_{L^2(\Omega)} \text{ and hence} \quad (4.16)$$

$$\|(\sigma, u)\|_U = \|\sigma\|_{H_{DIV}^1(\Omega)} + \|u\|_{H^1(\Omega)} \quad (4.17)$$

$$\text{CORRECT?!} \quad (4.18)$$

we can then check that all above terms in $J([\sigma, u], 0)$ are well defined. If we additionally assume $f(u) \in L^2(\Omega)$ for all u , the problem remains to be well-posed. Theoretically one could potentially only require $f \in H^{-1}(\Omega)$ but for the scope of this thesis, we will restrict ourselves to the former. Especially as we will later on have to require f to be twice differentiable to actually solve the problem numerically, [see Newton's method, section ...].

what do we get on \mathcal{A} with respect to what space? bounded? in previous section would have the bounds from X , how to do this here? how does this even make sense, hilbert space, strong form?

Another point that has not been mentioned so far but will have to be taken into account is the way of how 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 one entails the additional definition of an appropriate norm on the boundary while also requiring the treatment of the additional term. Since we have assumed it to be at least L_2 -regular the appropriate conditions can directly be imposed as part of the discretised system which will be discussed in more detail in the implementation section.

4.3 Norm Equivalence

\mathcal{B} induces an inner product on V and subsequently a norm. It is clearly a symmetric bilinear form. We also have that $\mathcal{B}([\sigma, u], [\sigma, u]) \geq 0$ for all $[\sigma, u] \in V$, it only remains to show that $\mathcal{B}([\sigma, u], [\sigma, u]) = 0 \Rightarrow [\sigma, u] = 0$. Show positive definiteness :

We clearly have $\mathcal{B}([\sigma, u], [\sigma, u]) \geq 0$ for all $(\sigma, u) \in V$ since $J([\sigma, u], 0)$ is the sum of two squared L^2 -norms. It remains to show that

$$\mathcal{B}([\sigma, u], [\sigma, u]) = 0 \iff (\sigma, u) = 0 \quad (4.19)$$

Can't be right, the point is that J is zero at the solution, so unless the solution is zero ...

In order to set a theoretical framework for the space induced by the scalarproduct \mathcal{B} we would like to establish norm equivalence comparable to the one in section 3.3 or in the paper of Z. Cai et al. on "First-order system least squares for second-order partial differential equations: Part I", [23], where they show ellipticity of their derived functional with respect to their *solution* space. That is we would like to know if there exists constants $\alpha, \beta > 0$ such that

$$\alpha \|(\sigma, u)\|_U \leq J([\sigma, u], 0) \leq \beta \|(\sigma, u)\|_U \quad \forall [\sigma, u] \in U. \quad (4.20)$$

The spaces differ from [23] in terms of the H_{DIV} definition, therefore we cannot apply their result. However it might still be possible to proceed similarly to them to show that these constants exists, it is however beyond the scope of thesis to fully check this but could be an interesting extension to the topic.

4.4 A Finite Element Space-Time Formulation

After having derived a continuous least squares formulation let us turn towards deriving a finite element discretisation of the problem. We want to consider conforming subspaces of U and W , hence we will be considering finite dimensional spaces $U_h \subset U$ and $W_h \subset W$ that are also defined on the entire space-time domain.

U_h contains the solution space for σ and u . They can be chosen independently from each other which can potentially be advantageous due to the different occurrences of their partial derivatives. Hence let us assume that $\sigma \in \tilde{V}_h$ and $u \in V_h$, where $U_h = \tilde{V}_h \times V_h$.

Suppose we have $\tilde{n} = \dim(\tilde{V}_h)$ and let $\{\tilde{\phi}_1, \dots, \tilde{\phi}_{\tilde{n}}\}$ be a basis of \tilde{V}_h and similarly $n = \dim(V_h)$ and $\text{span}\{\phi_1, \dots, \phi_n\} = V_h$. We furthermore assume V_h to be constructed such that $\inf_{u_h \in V_h} \|v - u_h\|_V \rightarrow 0$ as $h \rightarrow 0$ for all $v \in U$ and respectively the same for σ_h and \tilde{V}_h . It is also worth noting that since we are in a space-time setting we have $\tilde{\phi} = \tilde{\phi}(x, t)$ and $\phi = \phi(x, t)$.

We can then represent σ and u as a linear combination of basis functions in V_h or equivalently \tilde{V}_h , that is

$$\sigma_h(x, t) = \sum_{i=1}^{\tilde{n}} \sigma_i \tilde{\phi}_i(x, t) \quad u_h(x, t) = \sum_{i=1}^n u_i \phi_i(x, t) \quad (4.21)$$

The functional J then looks as follows

$$J(\sigma_h, u_h) = \left\| \sum_{i=1}^n u_i (\phi_i)_t - \sum_{i=1}^{\tilde{n}} \sigma_i \text{div}(\tilde{\phi}_i) - f \left(\sum_{i=1}^n u_i \phi_i \right) \right\|_Z^2 + \left\| \sum_{i=1}^{\tilde{n}} \sigma_i \tilde{\phi}_i - D(x) \nabla \left(\sum_{i=1}^n u_i \phi_i \right) \right\|_Z^2 \quad (4.22)$$

which will give rise to a vector-valued system once we discretise the domain.

In the arising variational formulation that we will also have to consider finite dimensional subspaces of W_h of W . In the scope of this thesis we restrict ourselves to the assumption that $W_h = U_h$. That is we introduce a set of test functions consisting of the basis vectors of \tilde{V}_h and V_h . That is the discretised weak form then reads

$$\text{Find } (\sigma_h, u_h) \in U_h \text{ such that } B \cdot [\sigma_h, u_h]^T = L_{\tilde{f}}(u_h) \quad (4.23)$$

where $B \in \mathbb{R}^{m \times m}$, with $m = \tilde{n} + n$, be the matrix arising from the discretised bilinear operator, and $L_{\tilde{f}}(u_h) \in \mathbb{R}^m$ being a discretised right-hand side. Since we assume that the solution $s_h = [\sigma_h, u_h]$ first contains all values corresponding to σ and then for u we obtain a block structure for B and $L_{\tilde{f}}$ of the following form.

$$B = \begin{bmatrix} B_{\sigma\sigma} & B_{\sigma u} \\ B_{u\sigma} & B_{uu} \end{bmatrix} \quad L_{\tilde{f}}(u_h) = \begin{bmatrix} (L_{\tilde{f}}(u_h))_{\sigma} \\ (L_{\tilde{f}}(u_h))_u \end{bmatrix} \quad (4.24)$$

Each entry of each of the blocks of B can be computed explicitly according to the subsequent schemes.

$$\text{For } B_{\sigma\sigma} : \quad B_{ij} = \langle \tilde{\phi}_j, \tilde{\phi}_i \rangle_Z + \langle \text{div}(\tilde{\phi}_j), \text{div}(\tilde{\phi}_i) \rangle_Z \quad \forall i, j \in \{1, \dots, \tilde{n}\} \quad (4.25)$$

$$\text{For } B_{\sigma u} : \quad B_{ij} = -\langle D(x) \nabla \phi_j, \tilde{\phi}_i \rangle_Z - \langle (\phi_j)_t, \text{div}(\tilde{\phi}_i) \rangle_Z \quad \forall i \in \{1, \dots, \tilde{n}\}, j \in \{\tilde{n} + 1, \dots, m\} \quad (4.26)$$

$$\text{For } B_{u\sigma} : \quad B_{ij} = -\langle D(x) \nabla \phi_i, \tilde{\phi}_j \rangle_Z - \langle (\phi_i)_t, \text{div}(\tilde{\phi}_j) \rangle_Z \quad i \in \{\tilde{n} + 1, \dots, m\}, \forall j \in \{1, \dots, \tilde{n}\} \quad (4.27)$$

$$\text{For } B_{uu} : \quad B_{ij} = \langle D(x) \nabla \phi_j, D(x) \nabla \phi_i \rangle_Z + \langle (\phi_j)_t, (\phi_i)_t \rangle_Z \quad \forall i, j \in \{\tilde{n} + 1, \dots, m\} \quad (4.28)$$

In the case that f is independent of u , that is $f = 0$ or $f = f(x, t)$, the derivative of f with respect to u , is clearly zero and therefore the right-hand side, which we will denote by

$L_{\tilde{f}} = L_{\tilde{f}}(u_h)$ to underline its independence of u only contains the following terms

$$(L_{\tilde{f}})_i = 0 \quad i \in \{1, \dots, \tilde{n}\} \quad (4.29)$$

$$(L_{\tilde{f}})_i = P\left(\sum_{j=1}^{\tilde{n}} \langle \operatorname{div}(\tilde{\phi}_j), f \rangle_Z\right)_i - \langle (\phi_i)_t, f \rangle_Z \quad i \in \{\tilde{n} + 1, \dots, m\} \quad (4.30)$$

where P is some sort of projection in case $\tilde{n} \neq n$, how to really write this up?

Thus we now know how compute each term we can assemble one large linear system of equations

$$B \begin{pmatrix} \sigma \\ u \end{pmatrix} = L_{\tilde{f}} \quad (4.31)$$

which can then be solved immediately for $[\sigma, u]^T$ using for example a multigrid method. In the case of $f = f(u)$ the situation is more complicated, since we also have to take the derivatives of f with respect to u into account, and then construct a nonlinear iteration scheme and consider linearisations of the problem to solve for u unless we have that $f(u)$ is a linear function. In that case we would only obtain additional terms that can be added to B_{uu} .

4.5 Computation of the Derivatives

In order to formulate a nonlinear iteration scheme we will first derive the first and second order Gateaux derivatives of J to be able to find concrete formulations for the gradient and hessian. We formulate them here now in their continuous form and will discuss the discretisation in the subsequent section. Furthermore for the remainder of this chapter we will assume that $\langle \cdot, \cdot \rangle = \langle \cdot, \cdot \rangle_Z$ in order to simplify the notation. For brevity and clarity we split the functional J in three different terms that will be considered separately dividing it into the linear and nonlinear terms which is possible due to the linearity of the inner product.

$$\begin{aligned} J_1(\sigma, u) &= \frac{1}{2} c_1 \langle u_t - \operatorname{div}(\sigma), u_t - \operatorname{div}(\sigma) \rangle \\ J_2(\sigma, u) &= \frac{1}{2} c_1 \langle 2u_t - 2\operatorname{div}(\sigma) - f(u), -f(u) \rangle \\ J_3(\sigma, u) &= \frac{1}{2} c_2 \langle \sigma - D(x)\nabla u, \sigma - D(x)\nabla u \rangle \end{aligned} \quad (4.32)$$

We can check that $J(\sigma, u) = J_1(\sigma, u) + J_2(\sigma, u) + J_3(\sigma, u)$.

Since $u \in H^1(\Omega)$ and $\sigma \in H_{DIV}(\Omega)$ they are not defined pointwise which we have to take into account for the subsequent computations.

What changes, what doesn't? Weak derivatives.

The partial directional derivatives of J_1 can be determined by once again taking the linearity

of the inner product as well as its symmetry into account

$$\begin{aligned}
\frac{\partial J_1}{\partial \sigma} &= \lim_{\epsilon \rightarrow 0} \frac{J_1(\sigma + \epsilon \tau, u) - J_1(\sigma, u)}{\epsilon} \\
&= \lim_{\epsilon \rightarrow 0} \frac{c_1}{2\epsilon} (\langle u_t - \operatorname{div}(\sigma + \epsilon \tau), u_t - \operatorname{div}(\sigma + \epsilon \tau) \rangle - \langle u_t - \operatorname{div}(\sigma), u_t - \operatorname{div}(\sigma) \rangle) \\
&= \lim_{\epsilon \rightarrow 0} \frac{c_1}{2\epsilon} (\langle u_t, u_t \rangle - \langle u_t, \operatorname{div}(\sigma) \rangle - \epsilon \langle u_t, \operatorname{div}(\tau) \rangle - \langle \operatorname{div}(\sigma), u_t \rangle + \langle \operatorname{div}(\sigma), \operatorname{div}(\sigma) \rangle \\
&\quad + \epsilon \langle \operatorname{div}(\sigma), \operatorname{div}(\tau) \rangle - \epsilon \langle \operatorname{div}(\tau), u_t \rangle + \epsilon \langle \operatorname{div}(\tau), \operatorname{div}(\sigma) \rangle + \epsilon^2 \langle \operatorname{div}(\tau), \operatorname{div}(\tau) \rangle \\
&\quad - \langle u_t, u_t \rangle + \langle u_t, \operatorname{div}(\sigma) \rangle + \langle \operatorname{div}(\sigma), u_t \rangle - \langle \operatorname{div}(\sigma), \operatorname{div}(\sigma) \rangle) \\
&= \lim_{\epsilon \rightarrow 0} \frac{c_1}{2\epsilon} (-2\epsilon \langle u_t, \operatorname{div}(\tau) \rangle + 2\epsilon \langle \operatorname{div}(\sigma), \operatorname{div}(\tau) \rangle + \epsilon^2 \langle \operatorname{div}(\tau), \operatorname{div}(\tau) \rangle) \\
&= -c_1 \langle u_t, \operatorname{div}(\tau) \rangle + c_1 \langle \operatorname{div}(\sigma), \operatorname{div}(\tau) \rangle
\end{aligned} \tag{4.33}$$

By proceeding analogously for equation J_2 and J_3 we obtain in these cases:

$$\frac{\partial J_2}{\partial \sigma} = c_1 \langle \operatorname{div}(\tau), f(u) \rangle \tag{4.34}$$

$$\frac{\partial J_3}{\partial \sigma} = c_2 \langle \sigma, \tau \rangle - c_2 \beta \langle \tau, \nabla u \rangle \tag{4.35}$$

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

$$\begin{aligned}
\frac{\partial J_1}{\partial u} &= \lim_{\epsilon \rightarrow 0} \frac{J_1(\sigma, u + \epsilon v) - J_1(\sigma, u)}{\epsilon} \\
&= \lim_{\epsilon \rightarrow 0} \frac{c_1}{2\epsilon} (\langle (u + \epsilon v)_t - \operatorname{div}(\sigma), (u + \epsilon v)_t - \operatorname{div}(\sigma) \rangle - \langle u_t - \operatorname{div}(\sigma), u_t - \operatorname{div}(\sigma) \rangle) \\
&= c_1 \langle u_t, v_t \rangle - c_1 \langle v_t, \operatorname{div}(\sigma) \rangle
\end{aligned} \tag{4.36}$$

$$\frac{\partial J_3}{\partial u} = -c_2 \langle \sigma, D(x) \nabla v \rangle + c_2 \langle D(x) \nabla u, D(x) \nabla v \rangle$$

Repeat assumptions on $D(x)$. But can leave it where it is for now?

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

$$\lim_{\epsilon \rightarrow 0} f(u + \epsilon v) = f(u) \text{ and} \tag{4.37}$$

$$\lim_{\epsilon \rightarrow 0} \langle f(u + \epsilon v), f(u + \epsilon v) \rangle - \langle f(u), f(u) \rangle = \langle f'(u) \cdot v, f(u) \rangle + \langle f(u), f'(u) \cdot v \rangle \tag{4.38}$$

which can be added due to symmetry.

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

$$(4.39)$$

Hence we obtain the following partial first order directional derivatives.

$$J_\sigma[\tau] = \frac{\partial}{\partial \sigma} J(\sigma, u)[\tau] = c_2 \langle \sigma, \tau \rangle + c_1 \langle \text{div}(\sigma), \text{div}(\tau) \rangle - c_2 \langle D(x) \nabla u, \tau \rangle - c_1 \langle u_t, \text{div}(\tau) \rangle - c_1 \langle f(u), \text{div}(\tau) \rangle \quad (4.40)$$

$$\begin{aligned} J_u[v] = \frac{\partial}{\partial u} J(\sigma, u)[v] &= c_1 \langle u_t, v_t \rangle - c_1 \langle v_t, \text{div}(\sigma) \rangle - c_2 \langle \sigma, D(x) \nabla v \rangle + c_2 \langle D(x) \nabla u, D(x) \nabla v \rangle \\ &\quad - c_1 \langle u_t, f'(u) \cdot v \rangle - c_1 \langle v_t, f(u) \rangle - c_1 \langle \text{div}(\sigma), f'(u) \cdot v \rangle + c_1 \langle f(u), f'(u) \cdot v \rangle \end{aligned} \quad (4.41)$$

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 .

$$\frac{\partial^2}{\partial \sigma^2} J[\tau][\rho] = c_2 \langle \rho, \tau \rangle + c_1 \langle \text{div}(\rho), \text{div}(\tau) \rangle \quad (4.42)$$

$$\frac{\partial^2}{\partial \sigma \partial u} J[\tau][v] = \frac{\partial^2}{\partial u \partial \sigma} J[\tau][v] = - \langle \tau, \nabla v \rangle - \langle v_t, \text{div}(\tau) \rangle - \langle \text{div}(\tau), f'(u) v \rangle \quad (4.43)$$

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

$$\begin{aligned} \frac{\partial^2 J}{\partial u^2} [v][w] &= c_1 \langle w_t, v_t \rangle + c_2 \langle \nabla w, \nabla v \rangle + c_1 \langle u_t, w^T f''(u) v \rangle - c_1 \langle w_t, f'(u) \cdot v \rangle - c_1 \langle v_t, f'(u) \cdot w \rangle \\ &\quad - c_1 \langle \text{div}(\sigma), w^T f''(u) v \rangle + c_1 \langle f(u), w^T f''(u) v \rangle + \langle f'(u) \cdot w, f'(u) \cdot v \rangle \end{aligned} \quad (4.44)$$

Yet some more on weak derivatives etc.

Thus we have now computed the necessary terms needed for the gradient as well as the Hessian of J . So let us now construct the nonlinear iteration scheme in order to minimise J .

4.6 Nonlinear Iteration Scheme

We only require the whole thing to be differentiable in a weak sense not like mentioned before.

As discussed in the previous chapter [see section 3.4] the general approach to finding a minimiser of the functional J is to search for a tuple $[\sigma, u]$ for which $\nabla J([\sigma, u]) = 0$, while $\nabla^2 J([\sigma, u])$ is positive definite. We start with an initial guess $s_0 = s_{\text{init}} = [\sigma_{\text{init}}, u_{\text{init}}]$ and then successively try to decrease energy. In case of a Newton step by finding a quadratic approximation of J at the value of the current iterate s_k , where the updated solution s_{k+1} is the minimiser of the quadratic approximation. However if J is not convex, and hence $\nabla^2 J([\sigma, u])$ not positive definite, the extremum of the quadratic approximation can actually lead us to a maximum of J . Therefore if we want to use a Newton iteration and maintain a decrease or at least no increase of energy in every step, that is $J(s_0) \geq J(s_1) \geq \dots \geq J(s_k) \geq \dots$ we need to be checking for convexity. In order to perform a Newton step we have to compute the Hessian of J , as well as its gradient in each iterate. The other option that was discussed previously to obtain a reduction in energy is to use a gradient descent method where we simply take an iteration step into the steepest descent direction. Hence we only need to evaluate the gradient in this case, which is computationally much cheaper but only leads to a linear convergence rate, instead of a quadratic one in the convex Newton case.

There are different ways to linearise and discretise in f , that is we have to decide how to represent f in our finite dimensional subspace formulation. For simplicity we assumed that $f \in C^2$, and denote the first and second derivative with respect to u by f' and f'' . This is in line with the forcing term stemming from the FitzHugh-Nagumo model as well as many other physical applications [source]. Hence for each degree of freedom in u we can determine coefficients for f, f', f'' and represent them in the basis of V_h , that is

$$(f_h)_i = \sum_{i=1}^n f_i \phi_i, \quad (f'_h)_i = \sum_{i=1}^n f'_i \phi_i, \quad (f''_h)_i = \sum_{i=1}^n f''_i \phi_i \quad (4.45)$$

Therefore f_h, f'_h and f''_h now represent a finite dimensional approximation of f for one particular approximation of u . With this representation we can now proceed to formulate a discretisation of the gradient of J , linearised in $s_k = [\sigma_k, u_k]$ where we know that inner products are still well-defined since we are working in conforming subspaces.

We want $\nabla J = 0$

$$(L_{\tilde{f}}(u_h))_\sigma = \langle \text{div}(\sigma_h), f'_h \rangle_Z \quad (4.46)$$

$$(L_{\tilde{f}}(u_h))_u = -\langle u_t, f'_h \rangle_Z - \langle v_t, f_h \rangle_Z + \langle f_h, f'_h \rangle_Z \quad (4.47)$$

$$\nabla J_k = \nabla J(s_k) = B \begin{pmatrix} \sigma_k \\ u_k \end{pmatrix} - L_{\tilde{f}}(u_k) \quad (4.48)$$

For one step of gradient descent we therefore compute the update by

$$s_{k+1} = s_k + \alpha(-\nabla J_k) \quad (4.49)$$

where $\alpha > 0$ is a scaling parameter that can be chosen in different ways, for example a line search algorithm and which will be discussed in more detail in the implementation chapter.

From the gradient and previous section we can determine the discretised Hessian which is needed for a Newton step. We obtain

$$H_k = \nabla^2 J(s_k) = B_{\text{lin}} + Q \quad (4.50)$$

where $B_{\text{lin}} = B$ from before and Q contains the nonlinear part, that is

$$Q = \begin{bmatrix} 0 & Q_{\sigma u} \\ Q_{u\sigma} & Q_{uu} \end{bmatrix} \text{ where} \quad (4.51)$$

$$(Q_{\sigma u})_{ij} = (Q_{u\sigma})_{ji} = -\langle \text{div}(\tilde{\phi}), f'_h \rangle \quad (4.52)$$

$$s_{k+1} = s_k - H_k^{-1}(\nabla J_k) \quad (4.53)$$

that is we would like to solve the linear system of equations

$$e_{k+1} = -H_k^{-1}(\nabla J_k) \quad \text{with} \quad s_{k+1} = s_k + e_{k+1} \quad (4.54)$$

But since it is generally expensive to compute the inverse of a large matrix, even if H_k is sparse and symmetric, because this property does in general not translate to the inverse we apply a multigrid method to solve (4.33). That is we need to solve a linear system of equations in each Newton step.

put derivative to zero therefore if we assume local convexity must be an extremum, positive definite, means minimum, therefore variational formulation makes sense

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)

A solution to (4.3) will always be a minimiser of the functional J independent on the choice of Z , however the sequence of iterates is dependent on the choice of the norm.

After having derived this discrete least squares space-time finite element formulation we will in the next chapter be looking at a concrete implementation which contains numerical results for some sample problems.