Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Diplom-Mathematikerin Helke Karen Hesse
aus Oberhausen

Tag der mündlichen Prüfung: 27. Juni 2008

Multiple Shooting and Mesh Adaptation for PDE Constrained Optimization Problems

Gutachter: Prof. Dr. Rolf Rannacher
Prof. Dr. Dr. h. c. Hans Georg Bock

Abstract

In this thesis, multiple shooting methods for optimization problems constrained by partial differential equations are developed, and, furthermore, a posteriori error estimates and local mesh refinement techniques for these problems are derived. Two different approaches, referred to as the direct and the indirect multiple shooting approach, are developed. While the first approach applies multiple shooting to the constraining equation and sets up the optimality system afterwards, in the latter approach multiple shooting is applied to the optimality system of the optimization problem. The setup of both multiple shooting methods in a function space setting and their discrete analogs are discussed, and different solution and preconditioning techniques are investigated. Furthermore, error representation formulas based on Galerkin orthogonality are derived. They involve sensitivity analysis by means of an adjoint problem and employ standard error representation on subintervals combined with additional projection errors at the shooting nodes. A posteriori error estimates and mesh refinement indicators are derived from this error representation. Several mesh structures originating from different restrictions to local refinement are discussed. Finally, numerical results for the solid state fuel ignition model are presented. This model describes an explosive system that does not allow the solution by standard solution techniques on the whole time domain and is a typical example for the application of time domain decomposition methods like multiple shooting.

Zusammenfassung

In dieser Doktorarbeit werden Multiple Shooting Verfahren für durch partielle Differentialgleichungen beschränkte Optimierungsprobleme entwickelt und zusätzlich a posteriori Fehlerschätzer und Methoden zur lokalen Gitterverfeinerung für diese Probleme ausgearbeitet. Es werden zwei unterschiedliche Ansätze, welche als direkter und indirekter Ansatz eines Multiple Shooting Verfahrens bezeichnet werden, betrachtet. Während der erste Ansatz das Multiple Shooting Verfahrens für die beschränkende Differentialgleichung ansetzt und anschließend das Optimalitätssystem aufstellt, wendet der letztere das Multiple Shooting Verfahren auf das Optimalitätssystem an. Die Darstellung beider Ansätze im Funktionenraum und die diskreten Entsprechungen werden diskutiert, und verschiedene Lösungs- und Vorkonditionierungstechniken werden untersucht. Des weiteren werden basierend auf Eigenschaften der Galerkinorthogonalität Fehlerdarstellungen hergeleitet. Diese beinhalten eine Sensitivitätsanalyse anhand von adjungierten Problemen und verwenden Fehlerdarstellungen auf Teilintervallen zusammen mit zusätzlichen Projektionsfehlern an den Zeitknoten des Multiple Shooting Verfahrens. Ausgehend von dieser Darstellung werden a posteriori Fehlerschätzer und Indikatoren für die Gitterverfeinerung hergeleitet. Verschiedene Gitterstrukturen, welche aus unterschiedlichen Restriktionen an die lokale Verfeinerung resultieren, werden diskutiert. Abschließend werden numerische Ergebnisse für ein Modell, welches die Zündungsphase eines Festkörperbrennstoffes beschreibt, angegeben. Dieses Modell beschreibt ein explosives Systems, das die Lösung mit Standardverfahren auf dem gesamten Zeitgebiet nicht zulässt, und das daher ein typisches Beispiel für die Anwendung von Zeitgebietszerlegungsmethoden, wie zum Beispiel Multiple Shooting Verfahren, darstellt.

Contents

1 Introduction

In this thesis, we develop and investigate *multiple shooting methods* for *optimal control problems constrained by parabolic partial differential equations*. Furthermore, we combine these multiple shooting methods with *a posteriori error estimation* techniques and *adaptive mesh refinement* procedures.

Systems of partial differential equations (PDEs) play an important role as models for dynamic processes, for example in physics, chemistry, biology, or engineering. Optimization problems occur as parameter estimation problems in the context of quantitative modeling or as optimal control or optimal design problems where a process has to be constructed or operated to meet certain objectives.

Reactors, built to study the details of chemical reactions, must provide stable and predictable environments (pressure, temperature, mixture of species) in order to avoid spurious observations. Therefore, the reactor must be controlled to maintain these environments. In dynamical processes, this should be achieved by optimal control, which can be interpreted as a constrained optimization problem. In this case, the constraints consist of a PDE initial boundary value problem and further technical restrictions. Thus, a typical example for optimal control problems constrained by PDEs with path and control constraints is the cost-minimal operation of a catalytic tube reactor under temperature restrictions [36] or the control of flow conditions for measurements in a high-temperature flow reactor [19]. Further examples of PDE constrained optimization problems are problems of catalytic reactions, for example the catalytic partial oxidation of methane in tubular reactors or the catalytic conversion of exhaust gas in passenger cars or a high-temperature flow reactor which has extensively been researched in [19].

Possible approaches to the solution of PDE constrained optimization problems are given by the class of *shooting methods*. Originally developed for the solution of *boundary value problems* (BVPs) in *ordinary differential equations* (ODEs), these approaches obtain their denomination from the typical solution process: For a guessed initial value, the approximation of the terminal time value is numerically calculated, and the approximation of the initial condition is improved by an iterative procedure. Metaphorically speaking, given an approximation of the initial value, we shoot onto the terminal time value and seek to match the prescribed value at this time point.

In general, we differentiate between *single shooting* and *multiple shooting methods*, though single shooting merely displays the special case of multiple shooting for one time interval as we will see later on.

Multiple shooting methods have proven to be the state-of-the-art for optimization problems in the context of ordinary differential or differential algebraic equation systems. Multiple

shooting methods simultaneously solve the constraints (the simulation or forward problem) and the optimization problem through globalized tailored infeasible Newton-like methods. They typically use time-adaptive strategies for the discretization of the differential equation constraints and tailored decomposition methods for the solution of the structured quadratic problems in every iteration ([9, 27]).

Multiple shooting methods possess several advantages. First, multiple shooting methods are *stable* and can be applied for the solution of highly instable problems. Second, the time domain decomposition allows the *introduction of knowledge about the process* at all timepoints by choosing adequate initial guesses for the states. Furthermore, multiple shooting methods allow the *parallel solution* of the subproblems on the different time subintervals.

The multiple shooting method as a *time domain decomposition* scheme goes back to the solution of *two point boundary value problems* for *ordinary differential equations* which are of interest not only in the context of optimization problems, but are often encountered in physics and engineering. Starting from the single shooting method, early developments into the direction of multiple shooting for two point boundary value problems can be found in the publication of Morrison, Riley, and Zancanaro [31], the article of Holt [23], and the article of Keller [26]. A good overview of the multiple shooting approach for ODE two point boundary value problems is given in the textbook of Stoer [39], where further extensions to ODE constrained optimization through the indirect approach are shortly introduced. This matter is also discussed in the report of Bulirsch [13]. The advantages of the direct approach are outlined in the diploma thesis of Plitt [33], in which first approaches to the software package MUSCOD are implemented and discussed, and further in the article of Bock and Plitt [11] and in the thesis of Bock [9]. Over the years a variety of different techniques for certain concrete ODE constrained problems has been derived from the original ideas of Bock, Holt, Keller, Plitt, and others and has been applied for the solution of those application problems mentioned above. In this context the solution of PDE constrained optimization problems by multiple shooting is brought down to the ODE approach by spatial discretization with the *method of lines* ([36]). For this reason the approach is limited to coarse spatial discretizations, and spatial mesh adaptation is not possible.

The idea to extend the multiple shooting approach for ODE constrained optimization to optimization problems which are constrained by parabolic PDEs is a rather new topic of research. First approaches were derived by Serban, Li, and Petzold who made first advances into the direction of adaptive mesh refinement in combination with multiple shooting. This approach is associated with the structured adaptive mesh refinement method (SAMR) and was first discussed in [37]. A direct multiple shooting approach for linear quadratic optimal control problems was developed by Ulbrich in [42] and Heinkenschloss in [20], and further extensions were presented by Comas in her doctoral thesis [14]. All these approaches are limited to linear quadratic optimal control problems and are mainly motivated by the possible parallelization of the intervalwise problems and by the reduced storage requirements. The efficient parallelization is difficult due to a lack of appropriate parallelizable preconditioners. The reduced storage requirements do not hold for the multiple shooting method in combination with adaptive mesh refinement obtained by the *dual weighted residual method* (DWR method). This restriction follows from the required storage of primal and dual variable over the whole time interval for the evaluation of the a posteriori error estimator.

Nevertheless, multiple shooting is of crucial importance for the solution of *highly instable constrained optimization problems* in which the constraining differential equation can not be solved for slightly disturbed control parameters. A typical example of an instable ODE constrained optimization problem is given by Example 1.1 below. We want to determine a time distributed control $q : I \to \mathbb{R}$ and a state function $u : I \to \mathbb{R}$ such that the cost functional (1.1a) is minimized while u fulfills the ordinary first order differential equation (1.1b). In this optimization problem we search the best possible approximation of $\bar{u} : I \to \mathbb{R}$ limited by a regularization term penalizing the control costs.

Example 1.1.

$$\min_{q,u} J(q,u) := \frac{\alpha}{2}\int_0^T |q(t)|^2 dt + \frac{1}{2}\int_0^T |u(t) - \bar{u}(t)|^2 dt \tag{1.1a}$$

such that

$$\begin{aligned} u' - \lambda e^u &= q \quad \text{on} \quad I = (0,T), \\ u(0) &= u_0. \end{aligned} \tag{1.1b}$$

For an appropriate choice of $\bar{u}$, the solution u of the constraining equation exists and is bounded for the optimal control q. But for the standard initial control $q = 0$ the solution of the constraining differential equation blows up: With the parameter $\lambda = 5$ and the initial value $u_0 = 0$ we search to calculate the solution for $q \equiv 0$. The solution has a blowup at about $t = 0.2$ as shown in Figure 1.1, and the numerical integration of the equation on the whole time interval is thus impossible.

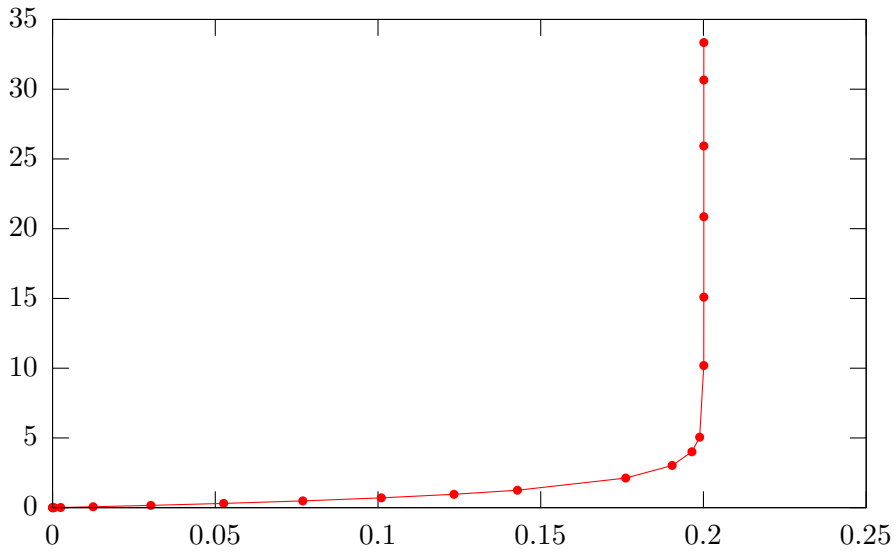


Figure 1.1: Behavior of $u(t)$ for different times $t \in [0, 0.25]$ for Example 1.1 with $\lambda = 5$ and $u_0 = 0$. (x-axis: t, y-axis: $u(t)$)

Multiple shooting as a time domain decomposition method on the other hand splits the time interval into small subintervals. This enables us to calculate intervalwise trajectories

and thereby to improve the intervalwise approximation of the control successively in the optimization process.

An analogous example can be stated for PDE constrained optimal control problems. We consider the *solid fuel ignition model* which in the context of optimization has been investigated in [25] or [24]. This problem is a powerful example for demonstrating the properties of *explosive systems*, and a comprehensive theoretical framework discussing the existence of solutions is available in the literature. We will describe these theoretical aspects and properties of the problem later on when considering a numerical application. For now, it is sufficient to present the pure problem formulation in Example 1.2 and to point out the motivation for applying multiple shooting techniques to this problem. Similar to the ODE example, we want to determine a control $q : I \to L^2(\Omega)$ as a source term on the right-hand side of the constraining equation (1.2b) such that the state $u : I \to L^2(\Omega)$ fulfills the constraining equation and approximates the given state $\bar{u} : I \to L^2(\Omega)$ in the cost functional (1.2a) best possible.

Example 1.2. (The solid fuel ignition model)

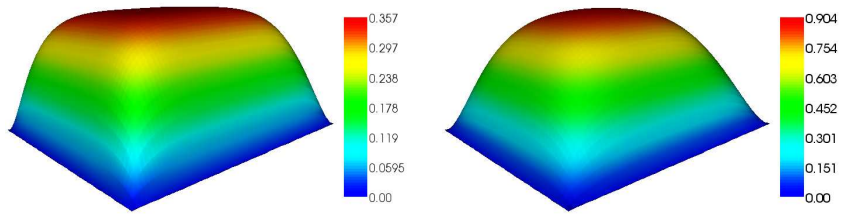
$$\min_{q,u} J(q,u) := \frac{\alpha}{2} \int_0^T \|q(t)\|^2 dt + \frac{1}{2} \int_0^T \|u(t) - \bar{u}(t)\|^2 dt \tag{1.2a}$$

subject to the constraining equation

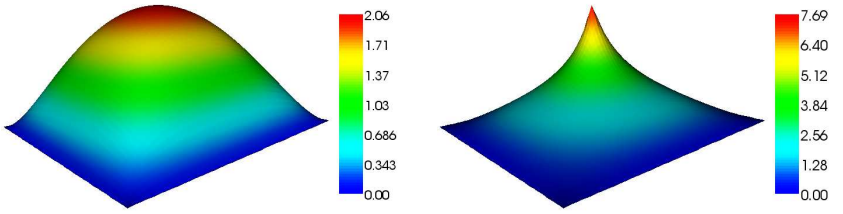
$$\begin{aligned} \partial_t u - \Delta u - \lambda e^u &= q \qquad \text{in} \quad I \times \Omega, \\ u(0) &= u_0 \qquad \text{in} \quad \Omega, \\ u(t,\cdot) &= 0 \qquad \text{on} \quad I \times \partial\Omega. \end{aligned} \tag{1.2b}$$

For the time interval $I = (0,1)$, the spatial domain $\Omega = (-1,1) \times (-1,1)$, the control $q \equiv 0$, the initial condition $u_0 = 0$, and the parameter $\lambda = 7.5$, the solution blows up at approximately $t = 0.14$. The solution on a 6 times globally refined mesh for different time points is shown in Figure 1.2. In contrast, the solution of the optimal control problem (we consider the simplest case of $\bar{u} \equiv 0$ in the following) is bounded in time. By application of multiple shooting, we are able to solve the intervalwise problems and obtain the correct solution after some steps of multiple shooting. We have chosen intervalwise constant controls in time and performed the calculation for 20 intervals with time step size 0.01. The control and corresponding solution obtained by the calculation are presented in Figure 1.3 and 1.4. The solution over the whole time interval does not only blow up for the easiest case of $q \equiv 0$, but also for several other tested initial controls. Therefore, the breakdown of standard optimization routines is likely, whereas multiple shooting with a sufficiently large number of intervals is suitable for the solution of the problem.

The accurate approximation of the solution to a PDE usually requires high computational effort which can be reduced by using adaptive grid strategies. Finite element schemes have proven to be very successful in this context. In particular, the method of *dual weighted residuals* (DWR method) is suited to speed up optimal control problems governed by partial differential equations, since it allows the efficient approximation of the goal of the optimization problem. Therefore, the combination of multiple shooting methods for PDE constrained optimization with mesh adaptation techniques is also discussed.

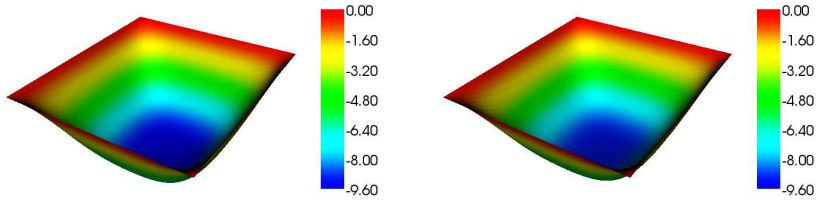


(a) $t = 0.04$ (b) $t = 0.08$

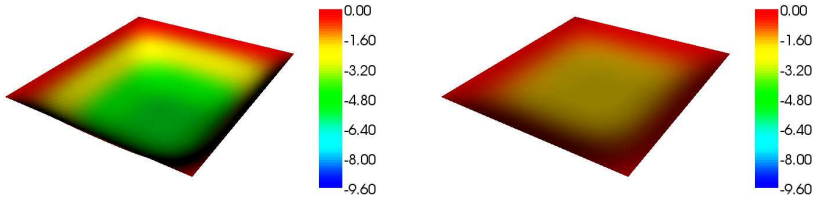


(c) $t = 0.12$ (d) $t = 0.1389$

Figure 1.2: Solution $u(t, x, y)$ for $q \equiv 0$, $u_0 = 0$, $\lambda = 7.5$ at different timepoints.



(a) $t = 0.25$ (b) $t = 0.75$

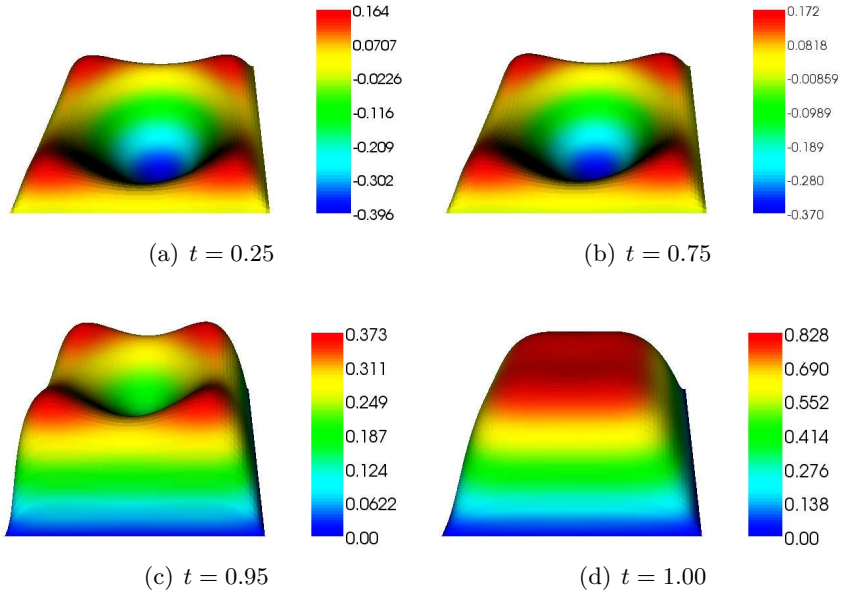


(c) $t = 0.95$ (d) $t = 1.00$

Figure 1.3: Control $q(t, x, y)$ for $\lambda = 7.5$, $\alpha = 10^{-2}$.

We now give a short overview of the topics related to mesh adaptive multiple shooting that are discussed in this thesis.

In Chapter 2 we present the **formulation and theory of PDE constrained optimization problems** and give a brief overview on the theoretical background concerning existence and uniqueness of solutions. Furthermore, we cite and prove needed standard results from the literature, for example first and second order optimality conditions.



(a) $t = 0.25$ (b) $t = 0.75$

(c) $t = 0.95$ (d) $t = 1.00$

Figure 1.4: Primal solution $u(t, x, y)$ for $\lambda = 7.5$, $\alpha = 10^{-2}$.

We proceed with an overview on the **historical background of the multiple shooting approach** in Chapter 3. Here, the historical motivation and development of multiple shooting for ODEs is summarized, and the insufficiency of the single shooting approach for the solution is discussed. We introduce multiple shooting and explain briefly further developments, such as condensing and efficient derivative generation. Furthermore, the basic idea of multiple shooting for ODE constrained optimization problems is briefly presented.

The idea of **multiple shooting for PDE constrained optimization** is introduced in Chapter 4. First, we develop the indirect multiple shooting approach, which applies multiple shooting techniques to the optimality system of the problem. After that, we introduce the direct multiple shooting approach which parameterizes the constraining equation and the cost functional by multiple shooting and derives the optimality system afterwards. Finally, we close the chapter with a theoretical investigation of the relation between direct and indirect multiple shooting.

Chapter 5 is devoted to the appropriate **discretization in time and space**. We introduce continuous Galerkin finite element methods on quadrilaterals as one possible method for the spatial discretization. For the time discretization, we present discontinuous Galerkin methods, and finally we discuss different possibilities for the discretization of the control space. Above all, the choice of the spatial meshes at the multiple shooting nodes is of great importance, and a variety of possible choices exists. Furthermore, the chapter provides us with two fundamentally different approaches for the choice of the changing spatial meshes in time. The first approach allows dynamically changing meshes in each time step, and the second approach is based on constant meshes for each subinterval. Finally, we present the implicit Euler time stepping scheme as the simplest example for a discontinuous Galerkin

time discretization scheme.

In Chapter 6 we discuss **solution techniques for the multiple shooting approach**. Newton's method is applied in both direct and indirect multiple shooting for solving the optimality system. The resulting linearized problem is solved by application of the generalized minimum residual method. We review different preconditioners and outline the necessity and efficiency of preconditioning by numerical examples. Additionally, in analogy to the ODE approach, we develop a condensing technique for direct multiple shooting which reduces the computational effort. The chapter closes with a numerical comparison of the different approaches.

Chapter 7 is devoted to the study of **a posteriori error estimation** for the discretization error of the cost functional. The discretization of state and control space is necessary for the computational solution of the problem and leads to inexact approximations of both control and state variables. This error results in an incorrect functional value, and the aim is to chose the discretization such that the error is minimal for a prescribed number of cells. As a first idea, the usual goal oriented dual weighted residual error estimator for PDE constrained optimization problems can be used as an add-on functionality after the solution of the problem by the multiple shooting approach. In the context of multiple shooting, this approach is limited to certain discretizations, where adjacent meshes on the multiple shooting nodes are the same. Therefore, we develop a new error estimation approach for the converged solution which allows the consideration of additional projection errors on the multiple shooting nodes.

In Chapter 8, we discuss different strategies for the combination of **multiple shooting and mesh adaptation**. First, we present the common idea of refinement due to the cellwise error indicators. This approach results in dynamically changing spatial meshes. Second, we develop a refinement strategy with intervalwise constant meshes, which equilibrates the projection error on the multiple shooting nodes with the discretization error on the intervals. Finally, numerical examples illustrate the efficiency of both approaches in comparison to global refinement.

In the context of **applications**, the **solid fuel ignition model** is a typical example for multiple shooting. We present this example in detail in Chapter 9. Here, the chemical and theoretical background are summarized. We discuss the reasons for the unstable behavior of the problem. Finally, we present results from numerical computations for different settings of the problem at the end of the chapter. All computations in this thesis were done with the finite element software package deal.II. The obtained solutions were visualized by means of the software VisuSimple.

The final Chapter 10 is devoted to an overview on multiple shooting for PDE constrained optimization, drawing **conclusions** and giving an **outlook** on further developments. The results obtained so far are summarized, and conclusions concerning the properties and applicability of the method are drawn. Possible extensions and promising future ways for developments of multiple shooting methods for PDE constrained optimization are briefly outlined.

2 Formulation and Theory of PDE Constrained Optimization Problems

This chapter covers a brief outline of the formulation and theory of PDE constrained optimization problems. We formulate an introductory example in order to develop the general and fundamental idea of parabolic optimization problems in Section 2.1 and continue with the usual mathematical setting and abstract formulation of these problems in Section 2.2. Well known results on existence and uniqueness of solutions to parabolic partial differential equations are briefly reviewed in Section 2.3, and the necessary and sufficient optimality conditions for parabolic optimization problems are revised in the final Section 2.4.

2.1 Preliminaries

Before we go into detail with respect to an abstract formulation of parabolic optimization problems, we give an introductory example to the kind of problems considered in this thesis. Our goal of optimization is finding an optimal control for a system governed by a partial differential equation of parabolic type. The simplest possible case is a linear problem involving the Laplacian with homogeneous Dirichlet boundary conditions and given initial value 0. The control parameter q is the source term of the equation. Utilizing an abstract cost functional $J(q,u)$ to be discussed later, our optimal control problem is: find a pair (q,u) in suitable spaces such that

Example 2.1.

$$J(q,u) = \min$$

under the constraints

$$\begin{aligned} \partial_t u + \Delta u &= q && \text{in} && I \times \Omega, \\ u &= 0 && \text{on} && I \times \partial\Omega, \\ u(0,.) &= 0 && \text{in} && \Omega \end{aligned}$$

in a polygonal domain $\Omega \subset \mathbb{R}^d$ and on a time interval $I = (0,T)$.

The next paragraph is devoted to the development of an abstract mathematical framework for parabolic optimization problems. Keeping the previous example in mind we generalize the formulation of the state equation. We present appropriate spaces for states and controls and specify the cost functionals of interest. Finally, we give further examples and embed them into the abstract formulation.

2.2 Formulation of Abstract Parabolic Optimization Problems

Let us first introduce the Hilbert spaces V and H, where V is continuously embedded and dense in H :

$$V \overset{d}{\hookrightarrow} H.$$

We identify the Hilbert space H with its dual space H^*, and together with the dual space of V, V^*, we retrieve the Gelfand triple

$$V \overset{d}{\hookrightarrow} H \cong H^* \overset{d}{\hookrightarrow} V^*.$$

Furthermore, the duality pairing of V^* and V is denoted by $\langle \cdot, \cdot \rangle_{V^* \times V}$, and the scalar product on H is given by $(\cdot, \cdot)_H$. In the following, we consider the continuous continuation of $(\cdot, \cdot)_H$ onto $\langle \cdot, \rangle_{V^* \times V}$ as a new representation for the functionals in V^*. This can be done due to the following remark:

Remark 2.1. Let the injection of V into V^* be denoted by $i : V \to H$. The dual mapping of i is the injection of H^* into V^* and is denoted by $i^* : H^* \to V^*$. From the definition of i^* the following identity holds for every $h \in H \cong H^*$:

$$\langle i^*(h), v \rangle_{V^* \times V} = (h, i(v))_H \quad \forall v \in V,$$

and we can consider h as a linear continuous functional on V. Due to the dense embedding of H^* into V^*, every functional $\langle v^*, \cdot \rangle_{V^* \times V}$ can be uniformly approximated by scalar products $(h, i(\cdot))_H$. Therefore it is reasonable to consider the continuous continuation of $(\cdot, \cdot)_H$ onto $\langle \cdot, \rangle_{V^* \times V}$ as a new representation for the functionals in V^*. A detailed description of this concept can be found for example in Lions [28] and Wloka [43].

Now, let R be a spatial Hilbert space for the control $q(t)$. We assume that the time dependent functions u and f have temporal values $u(t) \in V$ and $f(t) \in V^*$, and the initial value of our state u is given by $u(0) = u_0 \in H$. On a time interval $I = (0, T)$, $0 < T < \infty$, we consider *parabolic optimization problems* of the following abstract type:

$$\begin{aligned} \partial_t u(t) + A(u(t)) + B(q(t)) &= f(t), \\ u(0) &= u_0. \end{aligned} \tag{2.2}$$

Remark 2.2. (More general nonlinear equations) The decoupling of u and q in (2.2) is done for the purpose of notational simplification. The general case of an operator $C : X \times Q \to V^*$ with corresponding nonlinear PDE

$$\begin{aligned} \partial_t u(t) + C(u(t), q(t)) &= f(t), \\ u(0) &= u_0 \end{aligned}$$

can be treated analogously. Therefore, all results presented in this thesis can be applied to this case, too.

Here, B is assumed to be a (nonlinear) operator, with $B : R \to V^*$, given by a semi-linear form $\bar{b} : R \times V \to \mathbb{R}$ as

$$\langle B(\bar{q}), \bar{v} \rangle_{V^* \times V} = \bar{b}(\bar{q})(\bar{v}) \quad \forall \bar{v} \in V.$$

The elliptic spatial differential operator $A : V \to V^*$ is given in weak formulation by the semi-linear form $\bar{a} : V \times V \to \mathbb{R}$ as

$$\langle A(\bar{u}), \bar{v} \rangle_{V^* \times V} = \bar{a}(\bar{u})(\bar{v}) \quad \forall \bar{v} \in V.$$

For the weak formulation of problem (2.2) we introduce another Hilbert space X for the time dependent states,

$$X := W(I) := \{\, v \mid v \in L^2(I, V) \quad \text{and} \quad \partial_t v \in L^2(I, V^*) \,\},$$

for which we have (see, for example, [16]) a continuous embedding in $C(\bar{I}, H)$. Furthermore, we assume that the space Q of the controls is a subspace of $L^2(I, R)$,

$$Q \subseteq L^2(I, R).$$

Its scalar product and norm are denoted by $(\cdot, \cdot)_Q$ and $\|\cdot\|_Q$.

Now, we have the mathematical tools at hand to pose the state equation (2.2) in a weak form:

For a given control $q \in Q$ find a state $u \in X$ such that for all $\varphi \in X$

$$\int_I (\partial_t u(t), \varphi(t))_H dt + \int_I \bar{a}(u(t))(\varphi(t))dt + \int_I \bar{b}(q(t))(\varphi(t))dt = \int_I (f(t), \varphi(t))_H dt,$$
$$u(0) = u_0.$$

In the following, we omit the index H at the scalar product, $(\cdot, \cdot)$, and for the sake of brevity we additionally introduce the following notation:

$$((v, w)) := \int_I (v(t), w(t))dt,$$
$$a(u)(v) := \int_I \bar{a}(u(t))(v(t))dt,$$
$$b(q)(v) := \int_I \bar{b}(q(t))(v(t))dt.$$

By coupling the initial condition to the state equation, we retrieve by virtue of the abbreviatory notation the following compact form of the state equation:

$$((\partial_t u, \varphi)) + a(u)(\varphi) + b(q)(\varphi) + (u(0), \varphi(0)) = ((f, \varphi)) + (u_0, \varphi(0)) \quad \forall \varphi \in X. \tag{2.3}$$

The *objective* or *cost functional* of the optimization problem is denoted by $J : Q \times X \to \mathbb{R}$. We define J as the sum of two functionals, $J_1 : X \to \mathbb{R}$ and $J_2 : H \to \mathbb{R}$, and a regularization term by

$$J(q, u) = \alpha_1 J_1(u) + \alpha_2 J_2(u(T)) + \frac{\alpha_3}{2}\|q - \hat{q}\|_Q^2, \tag{2.4}$$

where we demand $\alpha_i \geq 0$, $i = 1, 2, 3$ and $\hat{q} \in Q$. Furthermore, we assume, that there is a functional $F : V \to \mathbb{R}$ such that

$$J_1(u) = \int_I F(u(t))dt. \tag{2.5}$$

We need this assumption for the consideration of the multiple shooting approach. In this context, we decompose the time domain I into smaller subintervals and want to consider the restriction of the cost functional J_1 to each of the subintervals.

Remark 2.3. Throughout this thesis, we set $\hat{q} = 0$ for the ease of presentation, but keep in mind that the general case follows straightforward and is often of relevance in application problems where a priori information on the control is available.

Remark 2.4. Later on, we mainly consider cost functionals of the following structure:

$$J_1(u) := \frac{1}{2}\int_I \|u(t) - \bar{u}(t)\|^2 dt \quad \text{and} \quad J_2(u(T)) := \frac{1}{2}\|u(T) - \bar{u}_T\|^2$$

where $\bar{u} \in X$ and $\bar{u}_T \in H$.

Remark 2.5. In the context of a posteriori error estimation, we assume that J_1 and J_2 are three times Gâteaux differentiable, which has to be verified for each concrete functional anew. In the case of J_1 and J_2 having the structure stated in Remark 2.4 this assumption clearly holds.

The goal of the optimization problem is now to minimize $J(q, u)$ under the constraining demand that q and u fulfill the state equation (2.3). Thus, the optimization problem reads

$$\min_{(q,u)\in Q\times X} J(q, u) \text{ subject to (2.3).} \tag{2.6}$$

Before discussing existence and uniqueness of solutions to parabolic optimization problems, let us first present three examples for problems of this type. We consider examples for two different types of quadratic functionals with linear and nonlinear constraining equations.

Let us first reconsider Example 2.1 in this abstract framework:

Example 2.2. (Distributed control of a terminal time functional) Let Ω be a bounded Lipschitz domain in $\mathbb{R}^d$. The optimal control problem is given by

$$\min_{(q,u)\in Q\times X} J(q, u) := \frac{\alpha_2}{2}\|u(T) - \bar{u}_T\|^2_{L^2(\Omega)} + \frac{\alpha_3}{2}\int_I \|q(t)\|^2_{L^2(\Omega)} dt$$

subject to the linear heat equation

$$\begin{aligned} \partial_t u - \Delta u &= q \quad && \text{in} \quad && \Omega \times I, \\ u &= 0 && \text{on} && \partial\Omega \times I, \\ u &= 0 && \text{in} && \Omega \times \{0\}. \end{aligned}$$

This example can be regarded in the previous abstract context by choosing the spaces

$$H = L^2(\Omega), \quad V = H_0^1(\Omega), \quad \text{and} \quad Q = L^2(I, L^2(\Omega)).$$

We have chosen $\alpha_1 = 0$, and J_2 is given by

$$J_2(u(T)) = \frac{1}{2}\|u(T) - \bar{u}_T\|^2_{L^2(\Omega)}.$$

The semi-linear forms are chosen as

$$\begin{aligned} a(u)(\varphi) &= ((\nabla u, \nabla\varphi)), \\ b(q)(\varphi) &= -((q, \varphi)). \end{aligned}$$

And finally the right-hand side and initial condition are given by

$$f = 0 \quad \text{and} \quad u_0 = 0.$$

Whereas in the previous example we wanted to fit a given function $\bar{u}_T$ at the terminal time point, we might furthermore be interested in matching a given time dependent function $\bar{u}(t)$:

Example 2.3. (Distributed control of a distributed functional)

$$\min_{(q,u)\in Q\times X} J(q,u) := \frac{\alpha_1}{2}\int_I \|u(t)-\bar{u}(t)\|^2_{L^2(\Omega)}dt + \frac{\alpha_3}{2}\int_I \|q(t)\|^2_{L^2(\Omega)}dt$$

subject to the nonlinear parabolic equation

$$\begin{aligned} \partial_t u - \Delta u + u^3 &= q && \text{in} && \Omega\times I,\\ u &= 0 && \text{on} && \partial\Omega\times I,\\ u &= 0 && \text{in} && \Omega\times\{0\}. \end{aligned}$$

For this example we have in the abstract formulation

$$H = L^2(\Omega),\quad V = H^1_0(\Omega),\quad \text{and}\quad Q = L^2(I,L^2(\Omega)).$$

With $\alpha_2 = 0$, the remaining part of the cost functional is given by

$$J_1(u) = \frac{1}{2}\int_I \|u(t)-\bar{u}(t)\|^2_{L^2(\Omega)}dt,$$

and the semi-linear forms are chosen as

$$\begin{aligned} a(u)(\varphi) &= ((\nabla u,\nabla\varphi)) + ((u^3,\varphi)),\\ b(q)(\varphi) &= -((q,\varphi)). \end{aligned}$$

And finally the right-hand side and initial condition are given by

$$f = 0\quad\text{and}\quad u_0 = 0.$$

Finally, we consider an example of Neumann boundary control:

Example 2.4. (Distributed Neumann boundary control of a distributed functional)

$$\min_{(q,u)\in Q\times X} J(q,u) := \frac{\alpha_1}{2}\int_I \|u(t)-\bar{u}(t)\|^2_{L^2(\Omega)}dt + \frac{\alpha_3}{2}\int_I \|q(t)\|^2_{L^2(\partial\Omega)}dt$$

subject to the nonlinear parabolic equation

$$\begin{aligned} \partial_t u - \Delta u + u^3 - u &= 0 && \text{in} && \Omega\times I,\\ \partial_n u &= q && \text{on} && \partial\Omega\times I,\\ u &= 0 && \text{in} && \Omega\times\{0\}. \end{aligned}$$

For this example we have in the abstract formulation

$$H = L^2(\Omega),\quad V = H^1_0(\Omega),\quad \text{and}\quad Q = L^2(I,L^2(\partial\Omega)).$$

Here, $\alpha_2 = 0$, and for J_1 we obtain

$$J_1(u) = \frac{1}{2}\int_I \|u(t)-\bar{u}(t)\|^2_{L^2(\Omega)}dt.$$

The semi-linear forms are chosen as

$$a(u)(\varphi) = (\!(\nabla u, \nabla \varphi)\!) + (\!(u^3, \varphi)\!),$$
$$b(q)(\varphi) = -\int_I (q, \varphi)_{L^2(\partial\Omega)} dt,$$

and finally the right-hand side and initial condition are given by

$$f = 0 \quad \text{and} \quad u_0 = 0.$$

We proceed with the discussion of existence and uniqueness of solutions.

2.3 Existence and Uniqueness of Solutions

The matter of existence and uniqueness of solutions to optimization problems as presented above has extensively been discussed for example in the textbooks of Lions [28], Fursikov [18], and Tröltzsch [41]. In the literature two different techniques are used for proving results on existence and uniqueness. On the one hand, the *reduced approach* is applied such that the states are considered as a function of the control q. On the other hand, the *non-reduced approach* treats the states and controls explicitly coupled. In the following, we refer to the reduced approach for the theoretical investigation of existence and uniqueness.

Let us first recall some abstract results on existence and uniqueness. We assume the existence of a solution operator $S : Q \to X$ which maps the control q onto the solution $u(q)$ of the constraining state equation (2.3). The validity of this assumption only depends on the unique solvability of the parabolic equation (2.3) and has to be verified for each problem in detail. Within the reduced approach, the reduced cost functional $j : Q \to \mathbb{R}$ is introduced as

$$j(q) := J(q, S(q)),$$

and the optimization problem (2.6) is reformulated as an unconstrained optimization problem

$$\min_{q \in Q} j(q), \quad q \in Q. \tag{2.10}$$

We apply the classical theorem on existence from the calculus of variations:

Theorem 2.1. *Let the reduced functional $j : Q \to \mathbb{R}$ be weakly lower semi-continuous, that is*

$$\liminf_{n\to\infty} j(q_n) \geq j(q) \quad \textit{whenever} \quad q_n \rightharpoonup q \quad \textit{in} \quad Q$$

and coercive over Q, that is

$$j(q) \geq \alpha \|q\|_Q + \beta$$

for every $q \in Q$ and for some $\alpha > 0$, $\beta \in \mathbb{R}$. Then problem (2.10) has at least one solution $q \in Q$.

Proof. See for example the textbook of Dacorogna [15]. □

Furthermore, for the uniqueness of the solution we have to demand stronger restrictions on the reduced functional j:

Theorem 2.2. *Let the reduced functional j fulfill the requirements of Theorem (2.1). If in addition j is strongly convex on Q, that is*

$$j(\lambda q_1 + (1-\lambda)q_2) < \lambda j(q_1) + (1-\lambda) j(q_2)$$

for all $\lambda \in (0,1)$ and all $q_1, q_2 \in Q$, $q_1 \neq q_2$, then problem (2.10) has a unique solution.

Proof. Let us assume that q_1 and q_2, $q_1 \neq q_2$, are solutions of (2.10). For $\lambda \in (0,1)$ the following inequality holds due to the strong convexity of j:

$$j(\lambda q_1 + (1-\lambda)q_2) < \lambda j(q_1) + (1-\lambda) j(q_2) = \min_{q \in Q} j(q).$$

This is in contradiction to the optimality of q_1 and q_2. □

For the application of these theorems to arbitrary nonlinear problems, the requirements on j have to be verified. We show unique solvability within the abstract framework of the previous section only for the simple case of Example 2.2.

Let us first state the unique solvability of the linear heat equation:

Theorem 2.3. *Let I be a bounded time interval and Ω be a bounded Lipschitz domain. Set $H = L^2(\Omega)$ and $V = H_0^1(\Omega)$. The linear parabolic equation*

$$\begin{aligned} \partial_t u - \Delta u &= f && \text{in} \quad \Omega \times I, \\ u &= 0 && \text{on} \quad \partial\Omega \times I, \\ u &= u_0 && \text{in} \quad \Omega \times \{0\} \end{aligned}$$

has a unique solution $u \in X$ for $f \in L^2(I, V^*)$ and $u_0 \in H$. Additionally, u depends continuously on the data:

$$(f, u_0) \mapsto u$$

is a continuous mapping from $L^2(I, V^*) \times H$ into X.

Proof. See for example the textbook of Lions [28]. □

Now, we can state the following theorem:

Theorem 2.4. *Let I be a bounded time interval and Ω be a bounded Lipschitz domain. Set $H = L^2(\Omega)$ and $V = H_0^1(\Omega)$, $Q = L^2(I, L^2(\Omega))$. Furthermore let $\alpha_1, \alpha_3 > 0$. Then the optimization problem*

$$\min_{(q,u) \in Q \times X} J(q,u) := \frac{\alpha_2}{2} \|u(T) - \bar{u}\|^2_{L^2(\Omega)} + \frac{\alpha_3}{2} \int_I \|q(t)\|^2_{L^2(\Omega)} dt \tag{2.11a}$$

subject to the linear heat equation

$$\begin{aligned} \partial_t u - \Delta u &= q \quad && in \quad && \Omega \times I, \\ u &= 0 && on && \partial\Omega \times I, \\ u &= 0 && in && \Omega \times \{0\} \end{aligned} \tag{2.11b}$$

has a unique solution $(q, u) \in Q \times X$.

Proof. From Theorem 2.3 the solution operator $S : Q \to X$, $Sq = u$ of equation (2.11b) is known to be continuous and linear. The continuity and convexity of the reduced cost functional $j : Q \to R$ can directly be seen from its definition

$$j(q) := J(q, S(q)) = \frac{\alpha_2}{2}\|S(q(T)) - \bar{u}\|^2_{L^2(\Omega)} + \frac{\alpha_3}{2}\int_I \|q(t)\|^2_{L^2(\Omega)} dt,$$

and thus it is weakly lower semi-continuous. Application of Theorem (2.1) yields the existence of at least one solution, and $\alpha_3 > 0$ ensures strong convexity of j and thus uniqueness of the solution. □

For more general, nonlinear parabolic optimization problems the procedure of proving existence and uniqueness of solutions is quite similar to the one presented for the case of linear quadratic optimal control problems. Nevertheless, the proofs are more complicated, and for further details we refer to the literature cited at the beginning of this section. Throughout this thesis, we assume that our optimization problem of interest (2.6) admits a (locally) unique solution. Furthermore, in the context of multiple shooting, we also assume that the intervalwise problems admit a (locally) unique solution.

2.4 Optimality Conditions

In this section, we present first order necessary and sufficient optimality conditions for problem (2.6) by means of the reduced approach. Before recalling appropriate theorems, we shortly review the standard definitions of differentiability in normed vector spaces.

Definition 2.1. (Directional derivative) Let X and Y be normed vector spaces and U be a neighborhood of a point $x \in X$, and let $f : U \to Y$. If for any $h \in X$ there exists the limit

$$f'(x)(h) := \lim_{t \searrow 0} \frac{f(x + th) - f(x)}{t},$$

then $f'(x)(h)$ is called the *directional derivative* of f at x in direction h. If this limit exists for all $h \in X$, then f is called *directionally differentiable* at x.

Definition 2.2. (Gâteaux derivative) Let X and Y be normed vector spaces and U be a neighborhood of a point $x \in X$, and let $f : U \to Y$ be directionally differentiable in x. If the directional derivative $f'(x)$ is a continuous linear mapping from X to Y, then f is called *Gâteaux differentiable* and $f'(x)$ is called the *Gâteaux derivative* of f at x.

Definition 2.3. (Fréchet derivative) Let X and Y be normed vector spaces and U be a neighborhood of a point $x \in X$, and let $f : U \to Y$. If there exists a continuous linear mapping $f'(x) : X \to Y$ such that

$$\lim_{\|h\|_X \to 0} \frac{\|f(x+h) - f(x) - f'(x)(h)\|_Y}{\|h\|_X} = 0,$$

then f is called *Fréchet differentiable* at x and $f'(x)$ is called the *Fréchet derivative* of f at x.

With these preparations at hand, we can state the first and second order necessary and second order sufficient optimality conditions. The theorems and proofs in a more detailed explanation are given in the book of Troeltzsch [41].

Theorem 2.5. *(First order necessary optimality condition) Let the reduced functional j be Gâteaux differentiable on an open subset $Q_0 \subseteq Q$. If $q \in Q_0$ is a local optimal solution of the optimization problem* (2.10)*, then q fulfills the first order necessary optimality condition*

$$j'(q)(\delta q) = 0 \quad \forall \delta q \in Q. \tag{2.12}$$

Proof. With Q_0 open and given direction $\delta q \in Q$ there exists due to local optimality of $q \in Q_0$ a positive $\lambda \in \mathbb{R}$ such that $q + \lambda \delta q \in Q_0$ and $j(q + \lambda \delta q) < j(q)$. Therefore, we have for the difference quotient

$$\frac{j(q + \lambda \delta q) - j(q)}{\lambda} \geq 0.$$

With $\lambda \searrow 0$ we obtain in the limit

$$j'(q)(\delta q) \geq 0.$$

Due to linearity of the Gâteaux derivative and with $-\delta q$ a feasible direction, we obtain analogously

$$j'(q)(\delta q) \leq 0$$

and thus the stated condition. □

Remark 2.6. (Additional convexity of j) In the special case that the functional j is additionally convex, that is for all $\lambda \in [0, 1]$ and all $q_1, q_2 \in Q$ there holds

$$j(\lambda q_1 + (1 - \lambda) q_2) \leq \lambda j(q_1) + (1 - \lambda) j(q_2),$$

condition (2.12) is not only a necessary but also a sufficient optimality condition of (2.10).

Theorem 2.6. *(Second order necessary optimality condition) Let the reduced functional j be two times continuously Fréchet differentiable on an open subset $Q_0 \subseteq Q$ of q. If $q \in Q_0$ is a local optimal solution of the optimization problem* (2.10)*, then it holds the second order necessary optimality condition*

$$j''(q)(\delta q, \delta q) \geq 0 \quad \forall \delta q \in Q.$$

Proof. With Q_0 open and given direction $\delta q \in Q$ there exists a positive $\lambda \in \mathbb{R}$ such that $q + \lambda\delta q \in Q_0$. From local optimality of q we obtain by Taylor expansion

$$0 \leq j(q + \lambda\delta q) - j(q) = \lambda j'(q)(\delta q) + \frac{\lambda^2}{2} j''(q)(\delta q, \delta q) + r_2^j(q, \lambda\delta q),$$

where r_2^j is a remainder term of second order. From the first order necessary optimality condition and division by $\lambda^2/2$ we obtain

$$0 \leq j''(q)(\delta q, \delta q) + \frac{2r_2^j(q, \lambda\delta q)}{\lambda^2}$$

and in the limit for $\lambda \searrow 0$

$$0 \leq j''(q)(\delta q, \delta q)$$

which completes the proof. □

Finally, we recall the second order sufficient optimality condition.

Theorem 2.7. *(Second order sufficient optimality condition) Let the reduced functional j be two times continuously Fréchet differentiable on a neighborhood $Q_0 \subseteq Q$ of q. Assume that q fulfills the first order necessary optimality condition* (2.12) *and that there exists a positive $\gamma \in \mathbb{R}$ such that the second order sufficient optimality condition holds:*

$$j''(q)(\delta q, \delta q) \geq \gamma \|\delta q\|_Q^2 \quad \forall \delta q \in Q.$$

Then there exists a positive constant $\rho \in \mathbb{R}$ such that the following quadratic growth condition holds:

$$j(q + \delta q) \geq j(q) + \frac{\gamma}{4}\|\delta q\|_Q^2$$

for all $\delta q \in Q$ with $\|\delta q\|_Q^2 \leq \rho$. Thus, q is a local solution of the optimization problem (2.10).

Proof. The proof is performed by application of Taylor expansion. With $\theta \in (0, 1)$ we obtain for ρ small enough such that $q + \delta q \in Q_0$

$$\begin{aligned} j(q + \delta q) &= j(q) + j'(q)(\delta q) + \frac{1}{2} j''(q + \theta\delta q)(\delta q, \delta q) \\ &= j(q) + \frac{1}{2} j''(q + \theta\delta q)(\delta q, \delta q) \\ &= j(q) + \frac{1}{2} j''(q)(\delta q, \delta q) + \frac{1}{2}[j''(q + \theta\delta q) - j''(q)](\delta q, \delta q). \end{aligned}$$

With the assumed continuity of j'' at hand, we retrieve for small $\|\delta q\|_Q \leq \rho$

$$\left[\|j''(q + \theta\delta q) - j''(q)\right](\delta q, \delta q)\| \leq \frac{\gamma}{2}\|\delta q\|_Q^2.$$

Inserting the second proposition of the theorem, we finally obtain the stated result

$$j(q + \delta q) \geq j(q) + \frac{\gamma}{2}\|\delta q\|_Q^2 - \frac{\gamma}{4}\|\delta q\|_Q^2 = j(q) + \frac{\gamma}{4}\|\delta q\|_Q^2.$$

□

These theoretical results have been presented rather for the sake of completeness than for the investigation of our problem of interest. Whenever necessary, we hint at the theorems in the actual context.

3 Historical Background of the Multiple Shooting Approach

This chapter presents a historical motivation of multiple shooting and gives an overview on certain properties of multiple shooting in the context of ODE optimization. We have seen in Chapter 1 that multiple shooting for PDE constrained optimization is amongst others motivated by the application to highly instable problems. The historical background, however, is due to the application to ODE boundary value problems. Therefore, in Section 3.1 we give a standard example of an ODE boundary value problem to illustrate the insufficiency of single shooting methods and summarize shortly the idea and application of multiple shooting to ODE boundary value problems in Section 3.2. The idea of condensing is presented in Section 3.3, and efficient derivative generation is briefly discussed in Section 3.4. Finally, in Section 3.5 the usual direct multiple shooting approach for ODE constrained optimization is introduced.

3.1 The Single Shooting Approach for ODE Boundary Value Problems

A common approach for the solution of boundary value problems in ODEs is the so called *single shooting* method. This approach suggests itself and is consequently of simple structure. It is introduced in most of the standard textbooks on ordinary differential equation numerics as the before mentioned book of Stoer and Bulirsch [39] from which we borrow the notation.

Remark 3.1. In the following, we make use of the classical ODE notation and furthermore of the common notation used in the context of single and multiple shooting for ODE problems. First of all, we denote the derivative with respect to time by $u'(t)$ and the solution of the corresponding differential equation by $u(t)$. The solution of an initial value problem which is depending on the initial value $s \in \mathbb{R}^n$, is written as $u(t; s)$.

In an ODE boundary value problem we want to find a function $u : (a, b) \to \mathbb{R}^n$ with

$$u'(t) = f(t, u(t)), \tag{3.1a}$$

where u and f consist of the components

$$u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{pmatrix}, \quad f(t, u(t)) = \begin{pmatrix} f_1(t, u_1(t), \ldots, u_n(t)) \\ \vdots \\ f_n(t, u_1(t), \ldots, u_n(t)) \end{pmatrix} \quad \text{for} \quad t \in (a, b),$$

such that with $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$,

$$r(u,v) = \begin{pmatrix} r_1(u_1,\ldots,u_n,v_1,\ldots,v_n) \\ \vdots \\ r_n(u_1,\ldots,u_n,v_1,\ldots,v_n) \end{pmatrix},$$

u fulfills the boundary condition

$$r(u(a),u(b)) = 0. \tag{3.1b}$$

Existence and uniqueness of solutions to problems of this type have extensively been studied in the literature mentioned in the introduction. We skip these theoretical aspects for the sake of brevity and go on with the solution of problem (3.1) by application of single shooting. The idea of the single shooting method is the reformulation of problem (3.1) as an initial value problem with an additional parameter $\sigma \in \mathbb{R}^n$ for the initial value. This parameter is determined iteratively during the solution process:

Find $\sigma = (\sigma_1,\ldots,\sigma_n) \in \mathbb{R}^n$ such that for u with

$$u'(t;\sigma) = f(t,u(t;\sigma)), \quad u(a;\sigma) = \sigma \tag{3.2a}$$

the following boundary condition holds:

$$F(\sigma) := r(\sigma,u(b;\sigma)) = 0. \tag{3.2b}$$

The equivalence of problem (3.2) and (3.1) is easily shown by elementary calculus and is not repeated in this overview. This problem of finding a zero σ of equation (3.2b) subject to the ODE initial value problem (3.2a) can for example be solved by application of Newton's method:

$$\sigma_0 \in \mathbb{R}^n, \quad \sigma_{k+1} := \sigma_k - DF(\sigma_k)^{-1}F(\sigma_k) \quad k = 0,1,\ldots.$$

This iteration needs the evaluation of the function and its derivative,

$$F(\sigma_k) \quad \text{and} \quad (DF(\sigma_k))_{ij} = \left(\frac{\partial F_i}{\partial \sigma_j}\right)(\sigma_k),$$

in each step. For the determination of $F(\sigma_k) = r(\sigma_k,u(b;\sigma_k))$ the initial value problem (3.2a) has to be solved with initial value $\sigma = \sigma_k$, and the calculation of the derivative can either be performed by straightforward application of difference quotients,

$$DF(\sigma_k) \approx \left(F^{\eta_1}(\sigma_k),\ldots,F^{\eta_n}(\sigma_k)\right),$$

where

$$F^{\eta_j}(\sigma_k) := \frac{F(\sigma_{1,k},\ldots,\sigma_{j,k}+\eta_j,\ldots,\sigma_{n,k}) - F(\sigma_{1,k},\ldots,\sigma_{j,k},\ldots,\sigma_{n,k})}{\eta_j},$$

or by more sophisticated derivative generation techniques. We go into this in more detail later on in this chapter.

However, the single shooting approach often lacks stability with respect to the solution of the initial value problem (3.2a). A standard example for this property can be found in the textbook of Stoer and Bulirsch [39] and shall not be repeated in this overview. Summarizing,

the example therein outlines that the computational solution of boundary value problems with single shooting is assorted with difficulties: for a general solution of an initial value problem of the type $u'(t) = f(t, u(t))$, $u(a; s) = s$ with Lipschitz continuous function f and Lipschitz constant $L > 0$, the following well known estimate on the sensitivity to errors in the initial data holds:

$$\|u(t; s_1) - u(t; s_2)\| \leq e^{L|t-a|}\|s_1 - s_2\|.$$

Obviously, the influence of the error can be bounded, when the interval of interest is chosen small enough. This property leads to the idea of multiple shooting for the solution of ODE boundary value problems.

3.2 The Direct Multiple Shooting Approach for ODE Boundary Value Problems

The *multiple shooting method* for the solution of *ODE two point boundary value problems* is based on the idea of solving initial value problems on small time subdomains in parallel. An outer iterative method is applied to match the intervalwise trajectories on the edges of the time subdomains. We reconsider the boundary value problem (3.1) for which a *time domain decomposition* of the interval is chosen

$$a = \tau_0 < \tau_2 < \ldots < \tau_m = b,$$

and $m+1$ additional variables, the *multiple shooting variables*, $s^0, \ldots, s^m \in \mathbb{R}^n$ are introduced. Now, consider the *intervalwise restricted initial value problems* which determine intervalwise functions $u^j : (\tau_j, \tau_{j+1}) \to \mathbb{R}^n$ with

$$(u^j)' = f(t, u^j), \quad u(\tau_j) = s^j, \quad j = 0, \ldots, m-1. \tag{3.3}$$

The solution u^j depends on the initial value s^j and is denoted by $u^j(\cdot; s^j)$ in the following. The goal of multiple shooting is the determination of the multiple shooting variables $s^0, \ldots, s^m$ such that the corresponding piecewise trajectories of (3.3) fit together at the edges of the intervals and the boundary condition is fulfilled.

In mathematical formulation: for $j = 0, \ldots, m$ find s^j such that

$$F(s) := \begin{pmatrix} F_0(s^0, s^1) \\ F_1(s^1, s^2) \\ \vdots \\ F_{m-1}(s^{m-1}, s^m) \\ F_m(s^0, s^m) \end{pmatrix} := \begin{pmatrix} s^1 - u^0(\tau_1; s^0) \\ s^2 - u^1(\tau_2; s^1) \\ \vdots \\ s^m - u^{m-1}(\tau_m; s^{m-1}) \\ r(s^0, s^m) \end{pmatrix} = 0 \tag{3.4}$$

with $u_j(\cdot; s^j)$ solution of (3.3), and $s := (s^0, \ldots, s^m)$. Newton type methods are appropriate choices for the solution of equation (3.4). For instance, the application of Newton's method yields the iteration

$$s_0 \in \mathbb{R}^n, \quad s_{k+1} := s_k - DF(s_k)^{-1}F(s_k) \quad \text{for} \quad k = 0, 1, \ldots.$$

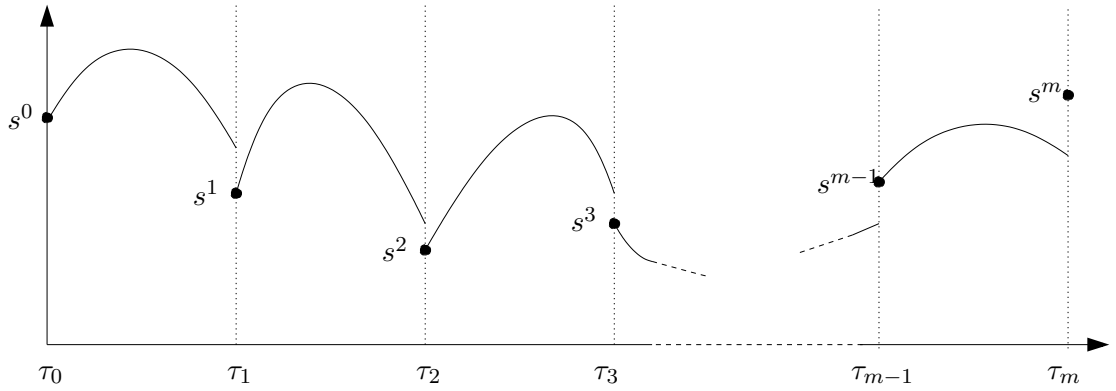


Figure 3.1: Idea of multiple shooting for boundary value problems.

With this basic description of the multiple shooting approach for the solution of two point boundary value problems at hand, we are able to present two important techniques for the efficient solution of the multiple shooting formulation of the problem. On the one hand, we are in need of solving the linearized system in Newton's method. On the other hand, most time of the multiple shooting method is spend in the assembling of the left hand side of the linear problem, which is equivalent to the derivative generation $\frac{\partial u^j}{\partial s^j}$ on each of the subintervals. Therefore, condensing techniques and efficient derivative generation play an important role in multiple shooting.

3.3 Condensing Techniques

This paragraph mainly equips us with the idea of *condensing* for the simplest case of multiple shooting. Nevertheless, these techniques can be extended to more complicated cases, especially to ODE constrained optimization problems. For the ease of presentation, we omit the index k of the Newton step in the sequel. The $n(m+1)\times n(m+1)$ Jacobian $DF(s) = (\frac{\partial F_i}{\partial s^j}(s))_{i,j=0,\dots,m}$ has, due to the special structure of the matching conditions, a sparse block structure of intervalwise Jacobians and identities,

$$DF(s) = \begin{pmatrix} -G_0 & I & 0 & & 0 \\ 0 & -G_1 & I & \ddots & \\ & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & -G_{m-1} & I \\ A & 0 & & 0 & B \end{pmatrix}.$$

Therefore, further simplifications can be performed on the system. Considering the blocks, the Jacobians $G_k, A, B \in \mathbb{R}^{n\times n}$ are determined by differentiation of the state equation and

the boundary condition with respect to the multiple shooting variables:

$$\begin{aligned} G_j &:= D_{s^j} F_j(s) = \frac{\partial u^j}{\partial s^j}(\tau_{j+1}; s^j), \quad j = 0, \dots m-1, \\ A &:= D_{s^0} F_m(s) = D_{s^0} r(s^0; s^m), \\ B &:= D_{s^m} F_m(s) = D_{s^m} r(s^0; s^m). \end{aligned}$$

If the Newton update for $s^j \in \mathbb{R}^n$ is denoted by $\Delta s^j \in \mathbb{R}^n$, $j = 0, \dots, m$, equation (3.4) turns into

$$\begin{aligned} G_0 \Delta s^0 - \Delta s^1 &= -F_0, \\ G_1 \Delta s^1 - \Delta s^2 &= -F_1, \\ &\vdots \\ G_{m-1} \Delta s^{m-1} - \Delta s^m &= -F_{m-1}, \\ A \Delta s^0 + B \Delta s^m &= -F_m. \end{aligned} \tag{3.5}$$

Simple transformation and recursive insertion of the equations yields

$$\begin{aligned} \Delta s^1 &= G_0 \Delta s^0 + F_0, \\ &\vdots \\ \Delta s^m &= G_{m-1} G_{m-2} \dots G_0 \Delta s^0 + \sum_{j=0}^{m-1} \left(\prod_{l=j+1}^{m-1} G_l \right) F_j. \end{aligned} \tag{3.6}$$

From the last identity of (3.5) we finally obtain the determining equation for the remaining increment Δs^0 as

$$(A + B G_{m-1} G_{m-2} \dots G_0) \Delta s^0 = w. \tag{3.7}$$

Thereby, the right hand side w is determined via

$$w = -(F_m + B F_{m-1} + B G_{m-1} F_{m-2} + \dots + B G_{m-1} G_{m-2} \dots G_1 F_0).$$

Hence, problem (3.5) for the evaluation of the Newton increment $\Delta s \in \mathbb{R}^{(m+1)\cdot n}$ is reduced to the solution of the linear system (3.7) for $\Delta s^0 \in \mathbb{R}^n$ and successive backward substitution according to (3.6) for the calculation of the remaining m increments. For further details on the convergence of Newton's method for this problem and the invertability of the matrix $A + B G_{m-1} G_{m-2} \dots G_0$ we refer to the literature mentioned in the introduction of this chapter.

It is easily verified by elementary calculus that in each Newton step m initial value problems have to be solved for the calculation of the residual, while for the explicit assembling of the left hand side matrix we need to solve $m \times n$ additional initial value problems – the assembling of G_j requires the solution of n initial value problems, one for each direction. Furthermore, to ensure convergence of Newton's method, a certain accuracy of the numerical solution must be guaranteed. Therefore, efficient techniques for the generation of the derivatives are indispensable. The next section is devoted to a brief review of two techniques that are commonly used. First, we discuss the (inefficient) application of difference quotients, and second, we present the internal numerical differentiation (IND) which is stable and highly efficient.

3.4 Derivative Generation

The first and simplest approach for the calculation of the derivatives is the application of difference quotients. In the context of the ODE example problem, this means to solve the initial value problem repeatedly with slightly perturbed initial values. This procedure results in a first order approximation of the derivative by finite differences. In detail, an approximation of the $n \times n$ Wronskian

$$W^j := \frac{\partial u^j}{\partial s^j}(\tau_{j+1}; s^j)$$

is obtained by perturbing the initial value in each component s_l^j, $l = 1, \ldots, n$, by the perturbation η_l, and performing this procedure for every component of the solution. Therefore, we define the perturbed vector as

$$s_{\eta_l}^j := (s_1^j, \ldots, s_{l-1}^j, s_l^j + \eta_l, s_{l+1}^j, \ldots, s_n^j)$$

and obtain the Wronskian as

$$W_{il}^j = (\frac{\partial u^j}{\partial s^j}(\tau_{j+1}; s^j))_{il} \approx (\frac{\partial u^j}{\partial s^j}^{\eta}(\tau_{j+1}; s^j))_{il} := \frac{u_i^j(\tau_{j+1}; s_{\eta_l}^j) - u_i^j(\tau_{j+1}; s^j)}{\eta_l}.$$

This method, sometimes also denoted as *external numerical differentiation*, has certain disadvantages. On the one hand the application of integration schemes with variable order and step size is difficult – slight perturbations in the initial value might lead to different integration steps, and a reliable calculation of the derivatives is not guaranteed. On the other hand, the application of integration schemes with fixed order and step size requires a much higher computational effort and is thus inefficient. The accuracy for the integration scheme would have to be set on a high level, and even then the best achievable accuracy for the calculation of the derivative is $\varepsilon_D = \sqrt{\varepsilon}$. Here ε denotes the accuracy to which the original trajectory is calculated. If we denote by

$$\left(\frac{\partial u^j}{\partial s^j}^{\eta,h}(\tau_{j+1}; s^j)\right)_{il}$$

the inexact approximation of the difference quotient obtained by numerical integration, we can write for the error

$$\varepsilon_D = \left(\frac{\partial u^j}{\partial s^j}^{\eta,h}(\tau_{j+1}; s^j)\right)_{il} - \left(\frac{\partial u}{\partial s^j}(\tau_{j+1}; s^j)\right)_{il} = \frac{\mathcal{O}(\varepsilon) + \mathcal{O}(\eta^2)}{\eta}.$$

A minimal bound of this expression is given $\varepsilon_D = \sqrt{\varepsilon}$ for $\eta^2 = \mathcal{O}(\varepsilon)$.

A promising alternative was first presented in [8]. The so called *internal numerical differentiation* (IND) is based on the idea of differentiating the discretization scheme itself. In the case of available exact derivatives of f (for example by automatic differentiation), this is equivalent to the solution of the variational differential equation with the integration scheme used for the solution of the original differential equation. Thus, the difference quotient

is approximated by the integrator with accuracy $\mathcal{O}(\varepsilon)$. Summarizing this, the achievable accuracy is for $\eta^2 = \mathcal{O}(\varepsilon_{\text{mach}})$ given by the identity

$$\varepsilon_D = \mathcal{O}(\varepsilon) + \mathcal{O}(\sqrt{\varepsilon_{\text{mach}}}),$$

where $\varepsilon_{\text{mach}}$ denotes the machine precision. Internal numerical differentiation allows us to reuse the step sizes and matrices of the integration scheme used for the calculation of $u^j(\tau_{j+1}; s^j)$. That is, the calculation of the nominal trajectory u^j and the corresponding Wronskian is performed simultaneously, which means that each time step is performed first for the nominal trajectory and then with the same parameters and matrix for the derivatives. This reduces, according to [9], the effort in comparison to external numerical differentiation by up to 80%.

3.5 The Multiple Shooting Approach for ODE Constrained Optimization Problems

The idea to apply multiple shooting techniques to ODE constrained optimization problems goes back to Bulirsch [13] where the indirect multiple shooting approach is developed for ODE constrained optimal control problems. A more popular approach, the direct approach, was introduced in the late seventies and early eighties, for example by Plitt [33] and Bock [11]. We give a short introduction to the direct approach, but do not go into detail concerning the solution of the multiple shooting problem. For details, we refer to the literature mentioned in the introduction.

Let $I = (0, T)$ denote the time interval, $u : I \to \mathbb{R}^{n_u}$ is the state variable, and $q : I \to \mathbb{R}^{n_q}$ is the control. Furthermore we define the cost functional $J : \mathbb{R}^{n_q} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_J}$ and the function $f : I \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_u}$. The optimization problem of interest reads

$$\min_{q,u} J(q, u) \tag{3.8a}$$

such that

$$u'(t) = f(t, u(t), q(t)), \quad u(0) = u_0. \tag{3.8b}$$

Most common approaches for the solution of this problem make use of a combination of iterative optimization tools and forward solvers for the constraining equation (3.8b). The multiple shooting approach, for example in [9], uses the idea of interpreting the problem (3.8) as a multi point boundary value problem which can be solved by multiple shooting. As before, the multiple shooting approach exploits the stable solution of smaller initial value problems on subintervals with additional matching conditions. The time domain decomposition and multiple shooting variables are chosen as before, $0 = \tau_0 < \tau_1 < \ldots \tau_{m-1} < \tau_m = T$ and $s^0, \ldots, s^m \in \mathbb{R}^{n_u}$, and the control is parameterized intervalwise by

$$q^j := q\Big|_{(\tau_j, \tau_{j+1})} \in \mathbb{R}^{n_q}, \quad j = 0, \ldots, m-1.$$

The intervalwise initial value problems are given by

$$u^{j'}(t) = f(t, u^j(t), q^j), \quad u(\tau_j) = s^j, \quad \text{for} \quad j = 0, \ldots, m-1.$$

Again, additional matching conditions have to be posed on the edges of the intervals to ensure continuity of the multiple shooting solution. These matching conditions are given by a system of nonlinear equations,

$$F(s^0, \dots, s^m, q^0, \dots q^{m-1}) = 0,$$

where the function F on the left hand side is defined as

$$F(s^0, \dots, s^m, q^0, \dots q^{m-1}) := \begin{pmatrix} s^1 - u^0(\tau_1; s^0, q^0) \\ s^2 - u^1(\tau_2; s^1, q^1) \\ \vdots \\ s^{m-1} - u^{m-2}(\tau_{m-1}; s^{m-2}, q^{m-2}) \\ s^m - u^{m-1}(\tau_m; s^{m-1}, q^{m-1}) \\ s^0 - u_0 \end{pmatrix}.$$

The reformulation of (3.8) in terms of the multiple shooting variables and the parameterized controls reads as follows:

$$\min_{s^0,\dots,s^m,q^0,\dots q^{m-1}} \tilde{J}(s^0, \dots, s^m, q^0, \dots q^{m-1}) \tag{3.9a}$$

such that the following equality constraints holds:

$$F(\tilde{s}, \tilde{q}) = 0. \tag{3.9b}$$

Here, the cost functional $\tilde{J}$ is obtained straightforward by inserting the parameterized states and controls into the original cost functional J. In [9], it has been shown that an appropriate choice of the multiple shooting nodes ensures the existence and boundedness of u^j on the intervals. Additionally, under appropriate assumptions on J and F, the well posedness and differentiability of problem (3.9) can be shown. Further on, the Lagrangian is set up and the optimality system has to be solved, usually by application of Newton type methods, like SQP methods and reduced SQP methods which are currently state of the art. Most approaches apply reduction and condensing techniques similar to equation (3.6) and (3.7) and efficient derivative generation techniques like backward differentiation formulas in combination with IND. For a description of Newton type methods for the solution of these problems, we refer to the textbook on numerical optimization of Nocedal and Wright [32], and a detailed description of derivative generation can be found in the diploma thesis of Albersmeyer [1].

Multiple shooting for ODE constrained optimization has been developed quite far during the last years. Nonlinear model predictive control problems as well as PDE constrained optimization problems have been solved by this approach, and a lot of effort has been put into the development of more efficient derivative generation techniques, condensing techniques and solvers, for example in [1], [35], or [10]. PDE constrained optimization problems have been solved by discretizing the PDEs with the method of lines and applying usual ODE multiple shooting techniques for the resulting system of ODEs. This approach is limited to fixed coarse spatial meshes such that the solutions lack spatial accuracy. Our approach will overcome these difficulties by applying multiple shooting directly to the PDE problem in function space. The next chapter introduces the idea of multiple shooting for PDE constrained optimization and provides an appropriate notational and theoretical framework.

4 The Multiple Shooting Approach for PDE Constrained Optimization

In this chapter we develop and investigate the basic ideas of multiple shooting for nonlinear optimal control problems constrained by partial differential equations. We start with the discussion of the general differences between multiple shooting for ODE constrained and PDE constrained optimization problems in Section 4.1. Afterwards, we proceed in Section 4.2 with a detailed development of the so called indirect multiple shooting approach for PDE constrained optimization problems. The chapter is closed by the description of the direct multiple shooting approach in Section 4.3. For both approaches, the notational and theoretical framework is derived, and the realization of the approach is outlined by means of simple example problems.

4.1 From ODEs to PDEs – Differences and Challenges

The generalization of multiple shooting approaches for ODE constrained to PDE constrained optimization problems assigns us with a variety of new challenges. The following composition describes the different changes and resulting tasks in developing a multiple shooting approach for PDE constrained optimization. For simplicity, we consider a system of only one component in the following listing.

1. First of all, in the case of PDE constrained optimization we have to consider not only a time dependent state, but in addition the *spatial dependence*. Whereas for ODEs with $f \in C(I)$ the solution is in $C^1(I)$, we now have to consider a solution in $W(I)$.

2. Consequently, for PDE constrained problems the multiple shooting variables are no longer in $\mathbb{R}$, but in $L^2(\Omega)$.

3. In the case of PDE constrained optimization, the *infinite dimensional spatial space* has to be *discretized* for the numerical solution.

4. For ODE constrained problems a high accuracy can easily be obtained by application of appropriate time stepping schemes of sufficiently high order. PDE constrained optimization requires *spatial mesh adaptation* in order to reduce the computational effort needed to obtain a certain accuracy.

5. For ODE problems, matrices of interest (for example for systems of equations) have a manageable number of entries, whereas *matrices* occurring in the context of PDEs usually tend to become quite *huge* due to a fine spatial discretization.

6. Spatial mesh adaptation possibly leads to *different adjacent meshes* at the multiple shooting nodes. Thus, we need an *appropriate formulation of the matching conditions*, for example by means of projection operators.

In the following sections, we present two possible multiple shooting approaches which overcome these difficulties. On the one hand, we consider the indirect multiple approach in which we apply multiple shooting to the optimality system of the original problem. On the other hand, the direct multiple shooting approach is presented which follows the classical ODE approach and parameterizes the constraining equation by multiple shooting before applying the Lagrange formalism on the cost functional and matching conditions. Summarizing, we differentiate between a first-optimize-then-multiple-shooting and a first-multiple-shooting-then-optimize approach. We will finally see how these approaches have a great deal in common but at the same time bear several varieties with respect to the performance.

4.2 The Indirect Multiple Shooting Approach

The indirect multiple shooting approach for PDE constrained optimization is a recent topic of research and was first introduced for linear quadratic optimal control problems in [21]. The derivation of this approach is given in the following. In contrast to the direct approach where multiple shooting is applied to the constraining equation, the basic idea of indirect multiple shooting is the application of multiple shooting to the optimality system of the optimization problem. Multiple shooting is applied to the primal and dual variable. The control is no longer part of the multiple shooting system, but is covered by the coupling of primal, dual, and control variable by boundary value problems with the same structure as the original optimality system.

Remark 4.1. In the following, we assume that the state equation and the intervalwise state equations have unique solutions. Furthermore, we request the unique solvability of all boundary value problems under consideration.

First, let us recall the constrained optimization problem (2.6):

The constraining parabolic partial differential equation is given by

$$(\!(\partial_t u, \varphi)\!) + a(u)(\varphi) + b(q)(\varphi) + (u(0), \varphi(0)) = (\!(f, \varphi)\!) + (u_0, \varphi(0)) \quad \forall \varphi \in X. \tag{4.1}$$

With J as defined in (2.4), the optimization problem of interest reads

$$\min_{(q,u)\in Q\times X} J(q,u) \text{ subject to (4.1)}. \tag{4.2}$$

Application of the Lagrange formalism yields the Lagrangian $\mathcal{L} : Q \times X \times X \to \mathbb{R}$ of the problem with Lagrange multiplier $z \in X$ which we refer to as the dual variable in the sequel.

$$\mathcal{L}(q,u,z) := J(q,u) - \left\{(\!(\partial_t u, z)\!) + a(u)(z) + b(q)(z) + (u(0), z(0)) - (\!(f, z)\!) - (u_0, z(0))\right\}. \tag{4.3}$$

By differentiation with respect to the states and the control, we retrieve the first order optimality system consisting of three equations, namely primal, dual and control equation:

Primal equation:

$$(\!(\partial_t u, \varphi)\!) + a(u)(\varphi) + b(q)(\varphi) + (u(0), \varphi(0)) = (\!(f, \varphi)\!) + (u_0, \varphi(0)) \quad \forall \varphi \in X. \tag{4.4a}$$

Dual equation:

$$-(\!(\partial_t z, \psi)\!) + a'_u(u)(\psi, z) + (z(T), \psi(T)) - J'_u(q, u)(\psi) = 0 \quad \forall \psi \in X. \tag{4.4b}$$

Control equation:

$$b'_q(q)(\chi, z) - J'_q(q, u)(\chi) = 0 \quad \forall \chi \in Q. \tag{4.4c}$$

Replacing J in (4.4) by the definition from equation (2.4), we get the following formulation:

Primal equation:

$$(\!(\partial_t u, \varphi)\!) + a(u)(\varphi) + b(q)(\varphi) + (u(0), \varphi(0)) = (\!(f, \varphi)\!) + (u_0, \varphi(0)) \quad \forall \varphi \in X. \tag{4.5a}$$

Dual equation:

$$-(\!(\partial_t z, \psi)\!) + a'_u(u)(\psi, z) + (z(T), \psi(T)) - \alpha_1 J'_1(u)(\psi) - \alpha_2 J'_2(u(T))(\psi(T)) = 0 \quad \forall \psi \in X. \tag{4.5b}$$

Control equation:

$$b'_q(q)(\chi, z) - \alpha_3 (q, \chi)_Q = 0 \quad \forall \chi \in Q. \tag{4.5c}$$

Multiple shooting is now applied to this optimality system as described in the sequel:

Let us, as in the approach for ordinary differential equations, decompose the time interval $I = (0, T)$ into m multiple shooting intervals $I_j := (\tau_j, \tau_{j+1})$ with

$$0 = \tau_0 < \tau_1 < \ldots < \tau_{m-1} < \tau_m = T.$$

For the purpose of multiple shooting in function space, we additionally introduce the intervalwise spaces

$$X^j := W(I_j) \quad \text{and} \quad Q^j := Q(I_j) := \left\{ q\Big|_{I_j} \;\middle|\; q \in Q \right\}$$

and the scalar products and norms $(\!(\cdot, \cdot)\!)_j$ on X^j, $(\cdot, \cdot)_{Q^j}$ and $\|\cdot\|_{Q^j}$ on Q^j. The intervalwise restriction of the states and controls u, z and q shall be denoted by

$$q^j := q\Big|_{I_j}, \quad u^j := u\Big|_{I_j}, \quad z^j := z\Big|_{I_j} \quad \text{for} \quad j = 0, \ldots, m-1.$$

Remark 4.2. (Choice of the control space) The problem formulation considered in this thesis in combination with intervalwise consideration of optimal control problems yields a restriction of the suitable control spaces. An equivalent reformulation of the original problem in the direct or indirect multiple shooting approach is only possible, if the intervalwise calculated control is in the control space Q, that is for arbitrary $q^0 \in Q^0, \ldots, q^{m-1} \in Q^{m-1}$ the composition $q : I \to R$ with $q\Big|_{I_j} = q^j$ fulfills the inclusion $q \in Q$. For clarification, consider the following concrete cases:

While for $Q = L^2(I, R)$ the stated requirement for Q is no restriction at all, we are not able to consider the case of a control which is constant on the whole time interval:

$$\begin{aligned} Q &= \left\{ v \in L^2(I,R) \;\middle|\; v(t) = c \in R \right\} \\ Q^j &= \left\{ v \in L^2(I_j,R) \;\middle|\; v(t) = c_j \in R \right\} \end{aligned} \tag{4.6}$$

and therefore $q^j = c_j$ but not necessarily $c_0 = c_1 = \ldots = c_{m-1} = c$. Consequently, the appropriate choice of Q in the context of multiple shooting is

$$Q(I) = \left\{ v \in L^2(I,R) \;\middle|\; v\big|_{I_j} = c_j \in R \right\}, \tag{4.7}$$

the space of intervalwise constant functions in time. Nevertheless, many control problems require a piecewise constant or piecewise linear control in time such that this restriction can be considered feasible.

On the multiple shooting nodes τ_j, $j = 0, \ldots, m$, we introduce the multiple shooting variables $s^j \in H$ as initial value for u^j in τ_j and $\lambda^{j+1} \in H$ as terminal time value for z^j in τ_{j+1}. Thus, we can now consider intervalwise two point boundary value problems of the same structure as (4.5) on each interval I_j. For the formulation of these problems, we introduce according to the notation in equation (2.5) the intervalwise functionals $J_1^j : X(I_j) \to \mathbb{R}$ by

$$J_1^j(u^j) := \int_{I_j} F(u(t))dt.$$

In this notation, the intervalwise boundary value problems read as follows:

Primal equation:

$$(\!(\partial_t u^j, \varphi)\!)_j + a(u^j)(\varphi) + b(q^j)(\varphi) + (u^j(\tau_j) - s^j, \varphi(\tau_j)) - (\!(f, \varphi)\!)_j = 0 \quad \forall \varphi \in X^j. \tag{4.8a}$$

Dual equation:

$$-(\!(\partial_t z^j, \psi)\!)_j + a'_u(u^j)(\psi, z^j) + (z^j(\tau_{j+1}) - \lambda^{j+1}, \psi(\tau_{j+1})) - \alpha_1 J_1^{j\prime}(u^j)(\psi) = 0 \quad \forall \psi \in X^j. \tag{4.8b}$$

Control equation:

$$b'_q(q^j)(\chi, z^j) - \alpha_3(q^j, \chi)_{Q^j} = 0 \quad \forall \chi \in Q^j. \tag{4.8c}$$

Next, we give the reformulation of the optimality system (4.5) in terms of the multiple shooting formulation. Therefore, we request additional matching conditions at the multiple shooting nodes and assume that the intervalwise states and controls solve (4.8):

Find $s^0, \ldots, s^m, \lambda^0, \ldots, \lambda^m$ such that

$$\begin{aligned} (s^0 - u_0, v) &= 0 \quad \forall v \in H, \\ (s^{j+1} - u^j(\tau_{j+1}), v) &= 0 \quad \forall v \in H, \quad j = 0, \ldots, m-1, \\ (\lambda^j - z^j(\tau_j), v) &= 0 \quad \forall v \in H, \quad j = 0, \ldots, m-1, \\ (\lambda^m, v) - \alpha_2 J_2'(s^m)(v) &= 0 \quad \forall v \in H. \end{aligned} \tag{4.9}$$

and q^j, u^j, z^j solve system (4.8) for $j = 0, \ldots, m-1$.

Remark 4.3. The introduction of s^0, λ^0, s^m and λ^m is artificial, and the variables could be removed from the system by inserting their definitions. Thus, it would be sufficient to introduce multiple shooting variables on the interior nodes. Nevertheless, their introduction allows us to handle the equations on all intervals identically, which turns out to be helpful with respect to the simplicity of the implementation.

Let us consider for $j = 0, \dots, m-1$ the solution operators

$$\Sigma_{u^j} : \begin{cases} H \times H \to X^j \\ (s^j, \lambda^{j+1}) \mapsto u^j \end{cases} \quad \text{and} \quad \Sigma_{z^j} : \begin{cases} H \times H \to X^j \\ (s^j, \lambda^{j+1}) \mapsto z^j \end{cases}$$

which map the boundary values s^j and λ^{j+1} onto the solutions u^j and z^j.

Furthermore we introduce the operators

$$\bar{\Sigma}_{u^j} : \begin{cases} H \times H \to H \\ (s^j, \lambda^{j+1}) \mapsto u^j(\tau_{j+1}) \end{cases} \quad \text{and} \quad \bar{\Sigma}_{z^j} : \begin{cases} H \times H \to H \\ (s^j, \lambda^{j+1}) \mapsto z^j(\tau_j) \end{cases}$$

mapping the boundary values s^j and λ^{j+1} onto $u^j(\tau_{j+1})$ and $z^j(\tau_j)$. We can now insert these operators into the system of matching conditions (4.9) and introduce the following notation: According to the number of multiple shooting variables in the indirect approach, we define the following abbreviations for the Cartesian product of vector spaces:

$$\tilde{H} := \underbrace{H \times \cdots \times H}_{m+1 \text{ times}}, \quad \tilde{Q} := Q^0 \times \cdots \times Q^{m-1}, \quad \tilde{X} := X^0 \times \cdots \times X^{m-1}. \tag{4.10a}$$

Furthermore we abbreviate the vectors of multiple shooting variables by

$$\begin{aligned} \tilde{s} &:= (s^0, \dots, s^m) \in \tilde{H}, & \tilde{u} &:= (u^0, \dots, u^{m-1}) \in \tilde{X}, \\ \tilde{\lambda} &:= (\lambda^0, \dots, \lambda^m) \in \tilde{H}, & \tilde{z} &:= (z^0, \dots, z^{m-1}) \in \tilde{X}, \\ \tilde{q} &:= (q^0, \dots, u^{q-1}) \in \tilde{Q} \end{aligned} \tag{4.10b}$$

and introduce on the product spaces the standard definition for the norm. For example, for $\tilde{H}$ this standard norm is given by

$$\|v\|_{\tilde{H}} := \sqrt{\sum_{i=0}^{m} \|v_i\|^2}.$$

From (4.9), we finally obtain the multiple shooting formulation: Find $(\tilde{s}, \tilde{\lambda}) \in \tilde{H} \times \tilde{H}$ such that

$$\begin{aligned} (s^0 - u_0, v) &= 0 \quad \forall v \in H, \\ (s^{j+1} - \bar{\Sigma}_{u^j}(s^j, \lambda^{j+1}), v) &= 0 \quad \forall v \in H, \quad j = 0, \dots, m-1, \\ (\lambda^j - \bar{\Sigma}_{z^j}(s^j, \lambda^{j+1}), v) &= 0 \quad \forall v \in H, \quad j = 0, \dots, m-1, \\ (\lambda^m, v) - \alpha_2 J_2'(s^{m-1})(v) &= 0 \quad \forall v \in H. \end{aligned} \tag{4.11}$$

This formulation is equivalent to the original problem (4.5) as stated in the following lemma.

Lemma 4.1. *Let $(q, u, z) \in Q \times X \times X$ be a solution to problem (4.5) and define for $j = 0, \dots, m-1$*

$$s^j := u(\tau_j), \quad \lambda^{j+1} := z(\tau_{j+1}). \tag{4.12}$$

Then $(\tilde{s}, \tilde{\lambda}) \in \tilde{H} \times \tilde{H}$ is a solution to problem (4.11).

Let $(\tilde{s}, \tilde{\lambda}) \in \tilde{H} \times \tilde{H}$ be a solution to problem (4.11) and let q^j be defined through the boundary value problem (4.8) for $j = 0, \dots, m-1$. Then $(q, u, z) \in Q \times X \times X$ defined by

$$\begin{aligned} u\big|_{I_j} &:= \Sigma_{u^j}(s^j, \lambda^{j+1}), \\ z\big|_{I_j} &:= \Sigma_{z^j}(s^j, \lambda^{j+1}), \quad j = 0, \dots, m-1, \\ q\big|_{I_j} &:= q^j, \quad j = 0, \dots, m-1 \end{aligned} \tag{4.13}$$

is a solution to problem (4.5).

Proof. Let $(q, u, z) \in Q \times X \times X$ be a solution to the boundary value problem (4.5) and for $j = 0, \dots, m-1$ let s^j, λ^{j+1} be defined through equation (4.12). Now let for s^j, λ^{j+1} on I_j the solutions of the boundary value problems (4.8) be given through q^j, u^j, z^j. For reasons of unique solvability of the boundary value problem (4.5) and its restriction (4.8) we directly retrieve the identities

$$q^j = q\Big|_{I_j}, \quad u^j = u\Big|_{I_j}, \quad z^j = z\Big|_{I_j}.$$

For reasons of continuity of q, u, and z we can easily see that s^j and λ^{j+1} and the corresponding solutions u^j, z^j fulfill the continuity conditions (4.11).

Now, let $(\tilde{s}, \tilde{\lambda}) \in \tilde{H} \times \tilde{H}$ be a solution to problem (4.11) and let $(q, u, z) \in Q \times X \times X$ be defined by the equations (4.13). Due to (4.11) q, u and z are continuous. By summing up the intervalwise boundary value problems (4.8) we can directly see that q, u and z are solutions of the boundary value problem (4.5). □

To make these rather abstract results easier understandable, we shortly present the indirect multiple shooting approach for Example 2.3.

Example 4.1. (Indirect multiple shooting for Example 2.3) The constraining equation is the heat equation with nonlinear reactive term u^3:

$$((\partial_t u, \varphi)) - ((\nabla u, \nabla \varphi)) + ((u^3, \varphi)) + (u(0) - 0, \varphi(0)) = ((q, \varphi)) \quad \forall \varphi \in X,$$

and in the cost functional we set $\alpha_1 = 1$, $\alpha_2 = 0$, $\alpha_3 = 1$. The spaces are defined as $H = L^2(\Omega)$, $V = H_0^1(\Omega)$, $R = L^2(\Omega)$ and $Q = L^2(I, L^2(\Omega))$. It is easily verified, that the optimality system for this problem reads as follows:

$$\begin{aligned} ((\partial_t u, \varphi)) + ((\nabla u, \nabla \varphi)) + ((u^3, \varphi)) + (u(0) - 0, \varphi(0)) &= ((q, \varphi)) && \forall \varphi \in X, \\ -((\partial_t z, \psi)) + ((\nabla z, \nabla \psi)) + ((3u^2 z, \psi)) + (z(T), \psi(T)) &= ((u - \bar{u}, \psi)) && \forall \psi \in X, \\ ((q, \chi)) &= -((z, \chi)) && \forall \chi \in X. \end{aligned} \tag{4.14}$$

We chose the time interval $I = (0, 1)$ and consider a multiple shooting time domain decomposition $0 = \tau_0 < \tau_1 < \tau_2 = 1$ of $m = 2$ multiple shooting intervals. In the framework

of multiple shooting for this problem we denote $\tilde{s} := (s^0, s^1, s^2)$ and $\tilde{\lambda} := (\lambda^0, \lambda^1, \lambda^2)$ and $\tilde{q} := (q^0, q^1)$ and retrieve the intervalwise boundary value problems for $j = 0, 1$:

For all $\varphi, \psi \in X^j$, $\chi \in Q^j$:

$$\begin{aligned} (\!(\partial_t u^j, \varphi)\!)_j + (\!(\nabla u^j, \nabla \varphi)\!)_j + (\!((u^j)^3, \varphi)\!)_j + (u^j(\tau_j) - s^j, \varphi(0)) &= (\!(q^j, \varphi)\!)_j, \\ -(\!(\partial_t z^j, \psi)\!)_j + (\!(\nabla z^j, \nabla \psi)\!)_j + (\!(3(u^j)^2 z^j, \psi)\!)_j + (z^j(\tau_{j+1}) - \lambda^{j+1}, \psi(T)) &= (\!(u^j - \bar{u}, \psi)\!)_j, \\ (\!(q^j, \chi)\!)_j &= -(\!(z^j, \chi)\!)_j. \end{aligned} \tag{4.15}$$

And finally the matching conditions for primal and dual solution are easily derived as

$$\begin{aligned} (s^0 - 0, \varphi) &= 0 \qquad \forall \varphi \in H, \\ (\lambda^0 - \bar{\Sigma}_{z^0}(s^0, \lambda^1), \varphi) := (\lambda^0 - z^0(\tau_0; s^1, \lambda^1), \varphi) &= 0 \qquad \forall \varphi \in H, \\ (s^1 - \bar{\Sigma}_{u^0}(s^0, \lambda^1), \varphi) := (s^1 - u^0(\tau_1; s^0, \lambda^1), \varphi) &= 0 \qquad \forall \varphi \in H, \\ (\lambda^1 - \bar{\Sigma}_{z^1}(s^1, \lambda^2), \varphi) := (\lambda^1 - z^1(\tau_1; s^1, \lambda^2), \varphi) &= 0 \qquad \forall \varphi \in H, \\ (s^2 - \bar{\Sigma}_{u^1}(s^1, \lambda^2), \varphi) := (s^2 - u^1(\tau_2; s^1, \lambda^2), \varphi) &= 0 \qquad \forall \varphi \in H, \\ (\lambda^2 - 0, \varphi) &= 0 \qquad \forall \varphi \in H. \end{aligned}$$

The optimization problem in terms of the indirect multiple shooting formulation requests the solution of the matching conditions (4.11) subject to the intervalwise boundary value problems (4.8). We discuss different techniques to solve this constrained problem of finding a zero to the system of matching conditions in Chapter 6. In what follows, we introduce and generalize the direct multiple shooting approach, which has first been addressed by Heinkenschloss and Comas in [20] and [14] for linear quadratic optimal control problems.

4.3 The Direct Multiple Shooting Approach

This section is devoted to development of the direct multiple shooting approach. We introduce an appropriate notational framework, which differs from the notation used in [14] and [20], and we generalize the approach presented in these publications to nonlinear optimization problems with arbitrary cost functionals. We consider again problem (4.2) and decompose the time interval $I = (0, T)$ into m multiple shooting intervals $I_j := (\tau_j, \tau_{j+1})$. As before we denote the intervalwise spaces by X^j and Q^j with the scalar products and norms $(\!(\cdot, \cdot)\!)_j$ on X^j, $(\cdot, \cdot)_{Q^j}$ and $\|\cdot\|_{Q^j}$ on Q^j. Furthermore, the intervalwise restrictions of the state u and control q are given by

$$q^j := q\big|_{I_j} \quad \text{and} \quad u^j := u\big|_{I_j} \quad \text{for} \quad j = 0, \ldots, m-1.$$

In contrast to the indirect approach, which reformulates the problem in terms of matching conditions and intervalwise boundary value problems, the direct multiple shooting approach is based on the formulation of intervalwise initial value problems with additional matching conditions. The transformations performed on the optimization problem (4.2) are similar

to the procedures needed for the direct multiple shooting approach for ODE constrained optimization.

We introduce on the multiple shooting node τ_j the multiple shooting variable $s^j \in H$ for $j = 0, \dots, m-1$ as initial value for the initial value problem (4.1) restricted to I_j:

$$((\partial_t u^j, \varphi))_j + a(u^j)(\varphi) + b(q^j)(\varphi) + (u^j(\tau_j), \varphi(\tau_j)) = ((f, \varphi))_j + (s^j, \varphi(\tau_j)) \quad \forall \varphi \in X^j. \tag{4.16}$$

The state variable u^j can now be treated as a function of the control q^j and the initial value s^j. We define on each interval the (nonlinear) solution operator S_j and the operator $\bar{S}_j$ which maps the initial value and the control onto the terminal time value $u(\tau_{j+1})$ of the solution:

$$S_j : \begin{cases} H \times Q^j \to X^j \\ (s^j, q^j) \mapsto u^j \end{cases} \quad \text{and} \quad \bar{S}_j : \begin{cases} H \times Q^j \to H \\ (s^j, q^j) \mapsto S_j(s^j, q^j)(\tau_{j+1}) \end{cases}.$$

For given q^j and s^j, $j = 0, \dots, m-1$, the state u^j is according to Remark 4.1 uniquely determined on I_j. By posing additional matching conditions on the edges of the intervals to ensure continuity of the global state, we are able to reformulate problem (4.2) as an equivalent equality constrained optimization problem in the multiple shooting variables $s^0, \dots, s^{m-1}, q^0, \dots, q^{m-1}$. We briefly recapitulate the abbreviatory notation for the vectors of multiple shooting variables and the Cartesian products of the spaces:

$$(\tilde{s}, \tilde{q}) := (s^0, \dots, s^{m-1}, q^0, \dots, q^{m-1}), \quad \tilde{H} := \underbrace{H \times \cdots \times H}_{m \text{ times}}, \quad \tilde{Q} := Q^0 \times \ldots \times Q^{m-1}.$$

In this notation, we define the reformulated cost functional $\bar{J} : \tilde{H} \times \tilde{Q} \to \mathbb{R}$ by

$$\bar{J}(\tilde{s}, \tilde{q}) := \sum_{j=0}^{m-1} \alpha_1 J_1^j(S_j(s^j, q^j)) + \alpha_2 J_2(\bar{S}_{m-1}(s^{m-1}, q^{m-1})) + \sum_{j=0}^{m-1} \frac{\alpha_3}{2} \|q^j\|_{Q^j}^2.$$

Now, an equivalent formulation of problem (4.2) in terms of the multiple shooting variables and parameterized controls reads

$$\min_{(\tilde{s}, \tilde{q}) \in \tilde{H} \times \tilde{Q}} \bar{J}(\tilde{s}, \tilde{q}) \tag{4.17a}$$

such that u^j solves (4.16) and the following matching conditions are fulfilled:

$$\begin{aligned} (s^0 - u_0, v) &= 0 \qquad \forall v \in H, \\ (s^{j+1} - \bar{S}_j(s^j, q^j), v) &= 0 \qquad \forall v \in H, \quad j = 0, \dots, m-2. \end{aligned} \tag{4.17b}$$

Remark 4.4. If $(\tilde{s}, \tilde{q}) \in \tilde{H} \times \tilde{Q}$ fulfills the equality constraints (4.17b) it is called a *feasible point* of problem (4.17). If we define for a feasible point the solutions $u^j := S_j(s^j, q^j) \in X^j$ for $j = 0, \dots, m-1$ and $(q, u) \in Q \times X$ with $q|_{I_j} := q^j$ and $u|_{I_j} := u^j$ then the following identity holds for the reformulated cost functional in (4.17a):

$$J(q, u) = \bar{J}(\tilde{s}, \tilde{q}).$$

This identity can directly be derived from the definition of $\bar{J}$.

Before proceeding with the description of the direct multiple shooting approach, let us state the equivalence of problem (4.2) and problem (4.17) in terms of the following lemma:

Lemma 4.2. *Let $(q,u) \in Q \times X$ be an optimal solution to problem (4.2). Then $(\tilde{s},\tilde{q}) \in \tilde{H} \times \tilde{Q}$, defined intervalwise by*

$$q^j := q\big|_{I_j} \quad \textit{and} \quad s^j := u(\tau_j) \quad \textit{for} \quad j = 0,\dots,m-1 \tag{4.18}$$

is an optimal solution to problem (4.17).

Let $(\tilde{s},\tilde{q}) \in \tilde{H} \times \tilde{Q}$ be an optimal solution to (4.17), then $(q,u) \in Q \times X$, defined by

$$q\big|_{I_j} := q^j \quad \textit{and} \quad u\big|_{I_j} := S_j(s^j,q^j) \quad \textit{for} \quad j = 0,\dots,m-1$$

is an optimal solution of problem (4.2).

Proof. We only prove the first part of the lemma, the proof of the second part is analogous. Assume $(q,u) \in Q \times X$ to be an optimal solution to problem (4.2) and let $q^j \in Q^j$ and $s^j \in H$, $j = 0,\dots m-1$, be defined by equation (4.18). Due to the unique solvability of equation (4.16) we can directly derive the identity $u\big|_{I_j} = S_j(s^j,q^j)$ for $j = 0,\dots,m-1$. For all $(\hat{q},\hat{u}) \in Q \times X$ we have due to the optimality of (q,u) the inequality

$$J(q,u) \leq J(\hat{q},\hat{u}). \tag{4.19}$$

Now suppose that there exists a feasible point $(\tilde{\delta s},\tilde{\delta q}) \in \tilde{H} \times \tilde{Q}$ with $\bar{J}(\tilde{\delta s},\tilde{\delta q}) < \bar{J}(\tilde{s},\tilde{q})$. Then $(\delta q,\delta u)$, defined by $\delta q\big|_{I_j} := \delta q^j$ and $\delta u^j := S_j(\delta s^j,\delta q^j)$ for $j = 0,\dots,m-1$ is a feasible point for problem (4.2). Thus, due to Remark 4.4, we can now write

$$J(\delta q,\delta u) = \bar{J}(\tilde{\delta s},\tilde{\delta q}) < \bar{J}(\tilde{s},\tilde{q}) = J(q,u),$$

which is in contradiction to assumption (4.19). □

The first order necessary optimality conditions for problem (4.17) are obtained by application of the Lagrange formalism. Let us introduce the Lagrange multipliers $p^j \in H$, $j = 0,\dots,m-1$, and define $\tilde{p} := (p^0,\dots,p^{m-1}) \in \tilde{H}$. With this notation, the Lagrangian $\mathcal{L} : \tilde{Q} \times \tilde{H} \times \tilde{H} \to \mathbb{R}$ of problem (4.17) is given by

$$\mathcal{L}(\tilde{q},\tilde{s},\tilde{p}) := \bar{J}(\tilde{q},\tilde{s}) - \left\{ (s^0 - u_0, p^0) + \sum_{j=0}^{m-2} (s^{j+1} - \bar{S}_j(s^j,q^j), p^{j+1}) \right\},$$

and the first order necessary optimality condition is obtained as the Karush Kuhn Tucker system (KKT system)

$$\mathcal{L}'(\tilde{q},\tilde{s},\tilde{p})(\tilde{\varphi},\tilde{\psi},\tilde{\chi}) = 0 \quad \forall (\tilde{\varphi},\tilde{\psi},\tilde{\chi}) \in \tilde{Q} \times \tilde{H} \times \tilde{H}.$$

Explicit calculation of the derivative on the left hand side yields a system of equations. We omit the arguments (s^j,q^j) of the derivatives of the solution operators for the purpose of notational brevity:

$$\begin{aligned} S'_{js^j}(\psi_j) &:= S'_{js^j}(s^j,q^j)(\psi_j), \qquad & S'_{jq^j}(\varphi_j) &:= S'_{jq^j}(s^j,q^j)(\varphi_j), \\ \bar{S}'_{js^j}(\psi_j) &:= \bar{S}'_{js^j}(s^j,q^j)(\psi_j), \qquad & \bar{S}'_{jq^j}(\varphi_j) &:= \bar{S}'_{jq^j}(s^j,q^j)(\varphi_j). \end{aligned}$$

With this notation we retrieve the following optimality system:

Differentiation with respect to p^j:

For $j = 0$ and for all $\varphi \in H$:

$$(s^0 - u_0, \varphi) = 0. \tag{4.20a}$$

For $j = 1, \dots, m-1$ and for all $\varphi \in H$:

$$(s^j - \bar{S}_{j-1}(s^{j-1}, q^{j-1}), \varphi) = 0. \tag{4.20b}$$

Differentiation with respect to s^j:

For $j = 0, \dots, m-2$ and for all $\psi \in H$:

$$\alpha_1 J_1^{j\prime}(u^j)(S'_{js^j}(\psi)) - (\psi, p^j) + (\bar{S}'_{js^j}(\psi), p^{j+1}) = 0. \tag{4.20c}$$

For $j = m-1$ and for all $\psi \in H$:

$$\alpha_1 J_1^{m-1\prime}(u^{m-1})(S'_{m-1s^{m-1}}(\psi)) + \alpha_2 J_2'(u^{m-1}(\tau_m))(\bar{S}'_{m-1s^{m-1}}(\psi)) - (\psi, p^{m-1}) = 0. \tag{4.20d}$$

Differentiation with respect to q^j:

For $j = 0, \dots, m-2$ and for all $\chi \in Q^j$:

$$\alpha_1 J_1^{j\prime}(u^j)(S'_{jq^j}(\chi)) + \alpha_3 (q^j, \chi)_{Q^j} + (\bar{S}'_{jq^j}(\chi), p^{j+1}) = 0. \tag{4.20e}$$

For $j = m-1$ and for all $\chi \in Q^{m-1}$:

$$\alpha_1 J_1^{j\prime}(u^{m-1})(S'_{m-1q^{m-1}}(\chi) + \alpha_2 J_2'(u^{m-1}(\tau_m))(\bar{S}'_{m-1q^{m-1}}(\chi)) + \alpha_3 (q^{m-1}, \chi)_{Q^{m-1}} = 0. \tag{4.20f}$$

For a better understanding let us briefly describe how the derivatives of the solution operators have to be interpreted in this formulation.

Lemma 4.3. *(Derivatives of the solution operators) Let for $j = 0, \dots, m-1$ the solution operators for the restricted problem* (4.16) *be given as before by*

$$S_j : \begin{cases} H \times Q^j \to X^j \\ (s^j, q^j) \mapsto u^j \end{cases} \quad \text{and} \quad \bar{S}_j : \begin{cases} H \times Q^j \to H \\ (s^j, q^j) \mapsto S_j(s^j, q^j)(\tau_{j+1}) \end{cases}.$$

Furthermore let the following problems for the determination of $v \in X^j$ and $w \in X^j$ on I_j be given:

$$((\partial_t v, \varphi))_j + a'_u(u^j)(v, \varphi) + (v(\tau_j) - \bar{s}, \varphi(\tau_j)) = 0 \quad \forall \varphi \in X^j, \tag{4.21}$$

$$((\partial_t w, \varphi))_j + a'_u(u^j)(w, \varphi) + b'_q(q^j)(\bar{q}, \varphi) + (w(\tau_j), \varphi(\tau_j)) = 0 \quad \forall \varphi \in X^j. \tag{4.22}$$

For the derivatives of the solution operators the following identities hold:

$$S'_{js^j} : \begin{cases} H \to X^j \\ \bar{s} \mapsto v \end{cases} \quad \text{and} \quad \bar{S}'_{js^j} : \begin{cases} H \to H \\ \bar{s} \mapsto v(\tau_{j+1}) \end{cases},$$

$$S'_{jq^j} : \begin{cases} Q^j \to X^j \\ \bar{q} \mapsto w \end{cases} \quad \text{and} \quad \bar{S}'_{jq^j} : \begin{cases} Q^j \to H \\ \bar{q} \mapsto w(\tau_{j+1}) \end{cases}.$$

Proof. The proof follows directly by differentiation of the restricted state equation (4.16) with respect to the initial value s^j and the restricted control q^j for $j = 0, \dots, m-1$. □

Remark 4.5. Lemma 4.3 yields that S'_{js^j} is the solution operator of problem (4.21) which is obtained by linearization of the original problem (4.16) with respect to the initial value into the direction $\bar{s}$. Therefore, the operator is mapping the initial value of the linearized problem, $\bar{s}$, to the solution v of equation (4.21). Furthermore, S'_{jq^j} is the solution operator of problem (4.22). Again, the equation is obtained as the linearization of (4.16), here with respect to the control into the direction $\bar{q}$. Thus, the solution operator maps $\bar{q}$ onto the solution w. Analogously, $\bar{S}'_{js^j}$ resp. $\bar{S}'_{jq^j}$ evaluate the solution v resp. w at the terminal time point τ_{j+1} of the interval I_j.

For further transformations of system (4.20), we additionally introduce adjoint operators and describe their interpretation as solution operators of the adjoint equations in Lemma 4.4.

$$\begin{aligned}
S'^*_{js^j} &:= S'^*_{js^j}(s^j, q^j) : X^{j*} \to H^*,\\
S'^*_{jq^j} &:= S'^*_{jq^j}(s^j, q^j) : X^{j*} \to Q^{j*},\\
\bar{S}'^*_{js^j} &:= \bar{S}'^*_{js^j}(s^j, q^j) : H^* \to H^*,\\
\bar{S}'^*_{jq^j} &:= \bar{S}'^*_{jq^j}(s^j, q^j) : H^* \to Q^{j*}.
\end{aligned}$$

We remind that, according to Remark 2.1, the Hilbert space H was identified with its dual space H^*.

Lemma 4.4. *For $j = 0, \dots, m-1$ let the solution operators for the equations* (4.21) *and* (4.22) *be given as before by*

$$S'_{js^j} : \begin{cases} H \to X^j \\ \bar{s} \mapsto v \end{cases} \quad \text{and} \quad \bar{S}'_{js^j} : \begin{cases} H \to H \\ \bar{s} \mapsto v(\tau_{j+1}) \end{cases},$$

$$S'_{jq^j} : \begin{cases} Q^j \to X^j \\ \bar{q} \mapsto w \end{cases} \quad \text{and} \quad \bar{S}'_{jq^j} : \begin{cases} Q^j \to H \\ \bar{q} \mapsto w(\tau_{j+1}) \end{cases}.$$

Furthermore let the following problems for the determination of $x \in X^j$ and $y \in X^j$ on I_j be given:

$$-((\partial_t x, \psi))_j + a'_u(u^j)(\psi, x) + (x(\tau_{j+1}), \psi(\tau_{j+1})) = \langle r, \psi \rangle_{X^{j*} \times X^j} \quad \forall \psi \in X^j, \tag{4.23}$$

$$-((\partial_t y, \psi))_j + a'_u(u^j)(\psi, y) + (y(\tau_{j+1}) - \zeta, \psi(\tau_{j+1})) = 0 \quad \forall \psi \in X^j. \tag{4.24}$$

For the adjoint operators the following identities hold:

$$\begin{aligned}
(S'^*_{js^j}(r), \xi) &= (x(\tau_j), \xi) && \forall \xi \in H,\\
\langle S'^*_{jq_j}(r), \chi \rangle_{Q^{j*} \times Q^j} &= -b'_q(q^j)(\chi, x) && \forall \chi \in Q^j,\\
(\bar{S}'^*_{js^j}(\zeta), \xi) &= (y(\tau_j), \xi) && \forall \xi \in H,\\
\langle \bar{S}'^*_{jq_j}(\zeta), \chi \rangle_{Q^{j*} \times Q^j} &= -b'_q(q^j)(\chi, y) && \forall \chi \in Q^j.
\end{aligned} \tag{4.25}$$

Proof. We prove only the first identity, the remaining equations can be shown analogously. By definition of the adjoint operator and the identification $H \cong H^*$, the following equation is true for $S'^*_{js^j}$:

$$(S'^*_{js^j}(r), \xi) = \langle r, S'_{js^j}(\xi)\rangle_{X^{j*}\times X^j} \quad \forall \xi \in H,\, \forall r \in X^{j*}. \tag{4.26}$$

From equation (4.23) we have for all $\xi \in H$:

$$\begin{aligned}\langle r, S'_{js^j}(\xi)\rangle_{X^{j*}\times X^j} &= -((\partial_t x, S'_{js^j}(\xi)))_j + a'_u(u^j)(S'_{js^j}(\xi), x) + (x(\tau_{j+1}), S'_{js^j}(\xi)(\tau_{j+1})) \\ &= ((\partial_t S'_{js^j}(\xi), x))_j + a'_u(u^j)(S'_{js^j}(\xi), x) + (x(\tau_j), S'_{js^j}(\xi)(\tau_j)).\end{aligned} \tag{4.27}$$

Equation (4.21) together with the definition of $S'_{js^j}(\xi)$ yields:

$$((\partial_t S'_{js^j}(\xi), x))_j + a'_u(u^j)(S'_{js^j}(\xi), x) + (x(\tau_j), S'_{js^j}(\xi)(\tau_j)) = (x(\tau_j), \xi) \quad \forall \xi \in H.$$

Now, together with equation (4.26) and (4.27) we retrieve the assertion

$$(S'^*_{js^j}(r), \xi) = (x(\tau_j), \xi) \quad \forall \xi \in H.$$

□

With these identities at hand, we are able to perform further simplifications of the KKT system (4.20). We prove the following theorem:

Theorem 4.5. *Let the dual and control equations* (4.20c), (4.20d), (4.20e) *and* (4.20f) *be given as before: Determine $\tilde{p} \in \tilde{X}^j$ and $\tilde{q} \in \tilde{Q}^j$ such that for all $\psi \in X^j$, $\chi \in Q^j$*

$$\alpha_1 J_1^{j\prime}(u^j)(S'_{js^j}(\psi)) - (\psi, p^j) + (\bar{S}'_{js^j}(\psi), p^{j+1}) = 0,\ j = 0, \dots, m-2, \tag{4.28a}$$

$$\alpha_1 J_1^{j\prime}(u^j)(S'_{js^j}(\psi)) + \alpha_2 J_2'(u^j(\tau_{j+1}))(\bar{S}'_{js^j}(\psi)) - (\psi, p^j) = 0,\ j = m-1, \tag{4.28b}$$

$$\alpha_1 J_1^{j\prime}(u^j)(S'_{jq^j}(\chi)) + \alpha_3(q^j, \chi)_{Q^j} + (\bar{S}'_{jq^j}(\chi), p^{j+1}) = 0,\ j = 0, \dots, m-2, \tag{4.28c}$$

$$\alpha_1 J_1^{j\prime}(u^j)(S'_{jq^j}(\chi)) + \alpha_2 J_2'(u^j(\tau_{j+1}))(\bar{S}'_{j,q_j}(\chi)) + \alpha_3(q^j, \chi)_{Q^j} = 0,\ j = m-1. \tag{4.28d}$$

An equivalent formulation of these equations is given by the system

$$\begin{aligned}(z^j(\tau_j) - p^j, \psi) = 0 \quad &\forall \psi \in H, \quad j = 0, \dots, m-1, \\ \alpha_3(q^j, \chi)_{Q^j} - b'_q(q^j)(\chi, z^j) = 0 \quad &\forall \chi \in Q^j, \quad j = 0, \dots, m-1\end{aligned}$$

where z^j is obtained as the solution of the following intervalwise initial value problems:

$$\begin{aligned}-((\partial_t z^j, \varphi))_j + a'_u(u^j)(\varphi, z^j) + (z^j(\tau_{j+1}) - p^{j+1}, \varphi(\tau_{j+1})) \\ = \alpha_1 J_1^{j\prime}(u^j)(\varphi) \quad \forall \varphi \in X^j, \quad j = 0, \dots, m-2,\end{aligned} \tag{4.29a}$$

$$\begin{aligned}-((\partial_t z^j, \varphi))_j + a'_u(u^j)(\varphi, z^j) + (z^j(\tau_{j+1}), \varphi(\tau_{j+1})) \\ = \alpha_1 J_1(u^j)^{j\prime}(\varphi) + \alpha_2 J_2'(u^j(\tau_{j+1}))(\varphi(\tau_{j+1})) \quad \forall \varphi \in X^j, \quad j = m-1.\end{aligned} \tag{4.29b}$$

Proof. The proof is mainly based on the interpretation of the adjoint operators (4.25). We transform the equations (4.28) by applying the definition of the adjoint operators such that the test functions are no longer arguments of the operators itself. The obtained equations allow the application of Lemma 4.4. The variables can be interpreted as solutions of the partial differential equations (4.23), (4.24), and the proposition follows by linear combination of the solutions. In detail, we proceed as follows:

Due to the assumptions on the parts of the functional in Remark 2.5 the linear functional $J_1'(u^j) : X^j \to \mathbb{R}$ is continuous and thus bounded on X^j. We write for a uniquely determined, but unknown, element $g_1 \in X^{j*}$ the representation

$$\alpha_1 J_1'(u^j)(v) = \langle g_1, v \rangle_{X^{j*} \times X^j} \quad \forall v \in X^j.$$

Accordingly, $J_2'(u^j(\tau_{j+1})) : H \to \mathbb{R}$ is a bounded linear functional on H, and we can write for an element $g_2 \in H$ the Riesz representation

$$\alpha_2 J_2'(u^{m-1}(\tau_m))(w) = (g_2, w) \quad \forall w \in H.$$

Now, we have everything at hand to transform equation (4.28) into a more compact formulation. First, we insert g_1 and g_2 into the equations. In the new formulation, we search $\tilde{p} \in \tilde{X}^j$ and $\tilde{q} \in \tilde{Q}^j$ such that for all $\psi \in H$, $\chi \in Q^j$

$$\begin{aligned}
\langle g_1, S'_{js^j}(\psi) \rangle_{X^{j*} \times X^j} - (p^j, \psi) + (p^{j+1}, \bar{S}'_{js^j}(\psi)) &= 0,\ j = 0, \dots, m-2, \\
\langle g_1, S'_{js^j}(\psi) \rangle_{X^{j*} \times X^j} + (g_2, \bar{S}'_{js^j}(\psi)) - (p^j, \psi) &= 0,\ j = m-1, \\
\langle g_1, S'_{jq^j}(\chi) \rangle_{X^{j*} \times X^j} + \alpha_3 (q^j, \chi)_{Q^j} + (p^{j+1}, \bar{S}'_{jq^j}(\chi)) &= 0,\ j = 0, \dots, m-2, \\
\langle g_1, S'_{jq^j}(\chi) \rangle_{X^{j*} \times X^j} + (g_2, \bar{S}'_{j,q_j}(\chi(\tau_m))) + \alpha_3 (q^j, \chi)_{Q^j} &= 0,\ j = m-1.
\end{aligned}$$

Second, we insert the definition of the adjoint operators and obtain the following system of equations:

$$(S'^*_{js^j}(g_1), \psi) - (p^j, \psi) + (\bar{S}'^*_{js^j}(p^{j+1}), \psi) = 0,\ j = 0, \dots, m-2, \quad (4.30a)$$

$$(S'^*_{js^j}(g_1), \psi) + (\bar{S}'^*_{js^j}(g_2), \psi) - (p^j, \psi) = 0,\ j = m-1, \quad (4.30b)$$

$$\langle S'^*_{jq^j}(g_1), \chi \rangle_{Q^{j*} \times Q^j} + \alpha_3 (q^j, \chi)_{Q^j} + \langle \bar{S}'^*_{jq^j}(p^{j+1}), \chi \rangle_{Q^{j*} \times Q^j} = 0,\ j = 0, \dots, m-2, \quad (4.30c)$$

$$\langle S'^*_{jq^j}(g_1), \chi \rangle_{Q^{j*} \times Q^j} + \langle \bar{S}'^*_{j,q_j}(g_2), \chi(\tau_m) \rangle_{Q^{j*} \times Q^j} + \alpha_3 (q^j, \chi)_{Q^j} = 0,\ j = m-1. \quad (4.30d)$$

We exploit (4.25) together with the linear partial differential equations (4.23) and (4.24). Introducing the intervalwise states $x^j, y^j \in X^j$ as solutions of the partial differential equations

$$\begin{aligned}
-(\!(\partial_t x^j, \varphi)\!)_j + a_u'(u^j)(\varphi, x^j) + (x^j(\tau_{j+1}), \varphi(\tau_{j+1})) &= \langle g_1, \varphi \rangle_{X^{j*} \times X^j} \\
&= \alpha_1 J_1^{j\prime}(u^j)(\varphi) \quad \forall \varphi \in X^j, \quad j = 0, \dots, m-1,
\end{aligned}$$

$$-(\!(\partial_t y^j, \varphi)\!)_j + a_u'(u^j)(\varphi, y^j) + (y^j(\tau_{j+1}) - p^{j+1}, \varphi(\tau_{j+1})) = 0 \quad \forall \varphi \in X^j, \quad j = 0, \dots, m-2,$$

$$
\begin{aligned}
-(\!(\partial_t y^j, \varphi)\!)_j + a'_u(u^j)(\varphi, y^j) + (y^j(\tau_{j+1}), \varphi(\tau_{j+1})) &= (g_2, \varphi(\tau_{j+1})) \\
&= \alpha_2 J_2'(u^j(\tau_{j+1}))(\varphi(\tau_{j+1}))) \quad \forall \varphi \in X^j, \quad j = m-1,
\end{aligned}
$$

the nonlinear system of equations (4.30) transforms into

$$
\begin{aligned}
(x^j(\tau_j), \psi) - (p^j, \psi) + (y^j(\tau_j), \psi) = 0 \qquad & \forall \psi \in H, \qquad j = 0, \dots, m-1, \\
-b'_q(q^j)(\chi, x^j) - b'_q(q^j)(\chi, y^j) + \alpha_3(q^j, \chi)_{Q^j} = 0 \qquad & \forall \psi \in Q^j, \qquad j = 0, \dots, m-1.
\end{aligned} \tag{4.32}
$$

Now, we can define $z^j := x^j + y^j \in X^j$ for $j = 0, \dots m-1$, and by linear combination of the linear PDEs for x^j and y^j we obtain the final formulation in terms of the dual variable z^j: Let z^j be determined by the partial differential equation

$$
\begin{aligned}
-(\!(\partial_t z^j, \varphi)\!)_j + a'_u(u^j)(\varphi, z^j) + (z^j(\tau_{j+1}) - p^{j+1}, \varphi(\tau_{j+1})) &= \langle g_1, \varphi \rangle_{X^{j*} \times X^j} \\
&= \alpha_1 J_1^{j\prime}(u^j)(\varphi) \quad \forall \varphi \in X^j, \quad j = 0, \dots, m-2,
\end{aligned}
$$

$$
\begin{aligned}
-(\!(\partial_t z^j, \varphi)\!)_j + a'_u(u^j)(\varphi, z^j) + (z^j(\tau_{j+1}), \varphi(\tau_{j+1})) &= \langle g_1, \varphi \rangle_{X^{j*} \times X^j} + (g_2, \varphi(\tau_{j+1})) \\
&= \alpha_1 J_1^{j\prime}(u^j)(\varphi) + \alpha_2 J_2'(u^j(\tau_{j+1}))(\varphi(\tau_{j+1}))) \quad \forall \varphi \in X^j, \quad j = m-1.
\end{aligned}
$$

The nonlinear system of equations (4.32) can be rewritten as

$$
\begin{aligned}
(z^j(\tau_j) - p^j, \psi) = 0 \qquad & \forall \psi \in H, \qquad j = 0, \dots, m-1, \\
\alpha_3(q^j, \chi)_{Q^j} - b'_q(q^j)(\chi, z^j) = 0 \qquad & \forall \psi \in Q^j, \qquad j = 0, \dots, m-1,
\end{aligned}
$$

which completes the proof. □

Remark 4.6. In order to symmetrize the notation for the primal and dual variables, we introduce an additional function $p^m \in H$ on the right boundary of the time interval. Furthermore, we remark, that the variable p^0 is obviously redundant. The matching conditions and defining equations for z^j transform into

$$
\begin{aligned}
(z^j(\tau_j) - p^j, \psi) = 0, \qquad & j = 1, \dots, m-1, \\
\alpha_2 J_2'(u^{m-1}(\tau_m))(\psi) - (p^m, \varphi) = 0, &
\end{aligned}
$$

where the dual variable $z^j \in X^j$ is obtained from the partial differential equation

$$
\begin{aligned}
-(\!(\partial_t z^j, \varphi)\!)_j + a'_u(u^j)(\varphi, z^j) + (z^j(\tau_{j+1}) - p^{j+1}, \varphi(\tau_{j+1})) &= (\!(g_1, \varphi)\!)_j \\
&= \alpha_1 J_1^{j\prime}(u^j)(\varphi) \quad \forall \varphi \in X^j, \quad j = 0, \dots, m-1.
\end{aligned} \tag{4.34}
$$

Next, we introduce solution operators for the dual equations. The variable z^j must be considered as a function not only of the control q^j, but also of the terminal time value p^{j+1} and via the dependence on u of s^j. The solution operators for the dual equation are defined by the following identities:

$$
\Xi_j : \begin{cases} H \times H \times Q^j \to X^j \\ (p^{j+1}, s^j, q^j) \mapsto z^j \end{cases} \quad \text{and} \quad \bar{\Xi}_j : \begin{cases} H \times H \times Q^j \to H \\ (p^{j+1}, s^j, q^j) \mapsto \Xi_j(p^{j+1}, s^j, q^j)(\tau_j) \end{cases}.
$$

Summarizing, we can write the multiple shooting formulation of the direct approach as follows. Inserting the reformulated matching conditions due to Theorem 4.5, considering Remark 4.6 and applying the definition of the solution operators, the direct multiple shooting formulation is given by the formulation: Find $\tilde{s} \in \tilde{H}$, $\tilde{p} \in \tilde{H}$, $\tilde{q} \in \tilde{Q}$ such that the following identities hold.

$$\begin{aligned}
(s^0 - u_0, v) &= 0 && \forall v \in H, && \\
(s^{j+1} - \bar{S}_j(s^j, q^j), v) &= 0 && \forall v \in H, && j = 0, \dots, m-2, \\
(p^j - \bar{\Xi}_j(p^{j+1}, s^j, q^j), \psi) &= 0 && \forall \psi \in H, && j = 1, \dots, m-1,. \\
(p^m, \varphi) - \alpha_2 J_2'(u^{m-1}(\tau_m))(\psi) &= 0 && \forall \psi \in H, && \\
\alpha_3 (q^j, \chi)_{Q^j} - b_q'(q^j)(\chi, \Xi_j(p^{j+1}, s^j, q^j)) &= 0 && \forall \chi \in Q^j, && j = 0, \dots, m-1.
\end{aligned} \tag{4.35}$$

Finally, by means of Theorem 4.5 we can point out the relation between the direct and the indirect multiple shooting approach. We have already seen that both approaches reformulate the original problem in terms of a (nonlinear) system of equations which allows the application of a nonlinear iterative solver as we will see in Chapter 6. These systems of equations for direct and indirect multiple shooting are closely related as is depicted in Figure 4.1. The equations

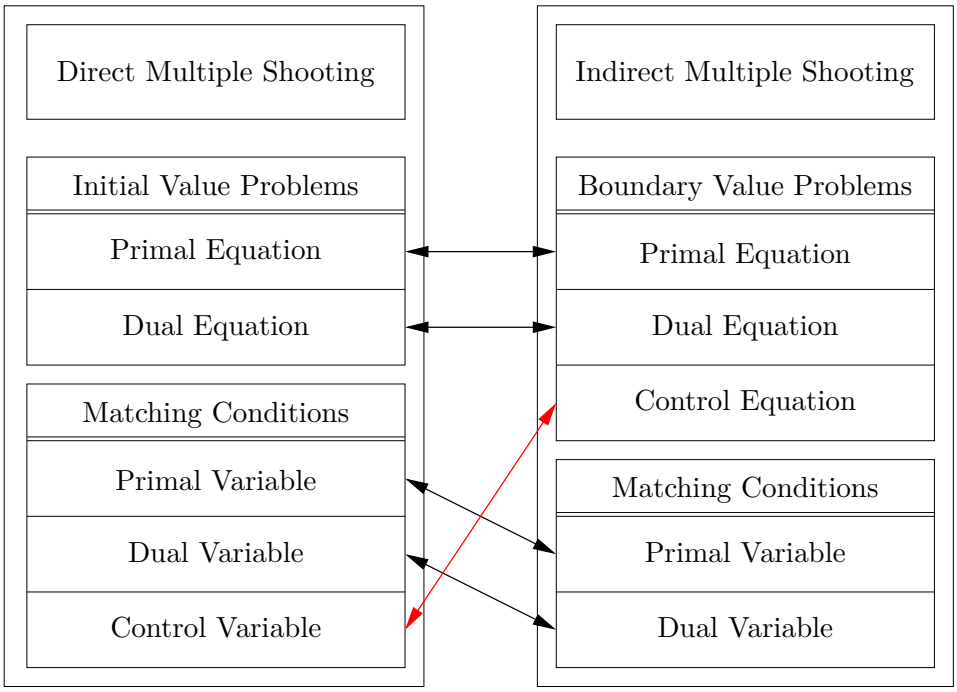


Figure 4.1: Relation between direct and indirect multiple shooting.

for primal and dual intervalwise states coincide for the direct and indirect multiple shooting approach. Keeping in mind the redundancy of λ_m and s_m due to Remark 4.3, this can be seen from equations (4.16), (4.34) and (4.8a), (4.8b). Additionally the matching conditions for the primal and dual variables coincide due to equations (4.35) and (4.9). Furthermore, the matching condition corresponding to the control in the direct case, the last equation of (4.35),

and the control equation (4.8c) are identical. Thus, we see that equations and matching conditions are closely related, and the essential difference is the elimination of the control from the system of matching conditions in the indirect multiple shooting approach. This yields intervalwise feasible states and controls due to the intervalwise solution of boundary value problems. These boundary value problems can be interpreted as optimality systems of intervalwise optimal control problems. In contrast, for the direct approach feasibility is obtained not until convergence of the multiple shooting approach. We see the advantages and disadvantages of both of these properties later on, when discussing the numerical solution of the multiple shooting formulations.

Example 4.2. (Direct multiple shooting for Example 2.3) For a better understanding of this abstract formulation of the direct multiple shooting approach, let us now take a look at the case of Example 2.3 with $\alpha_1 = 1$, $\alpha_2 = 0$, and $\alpha_3 = 1$. We chose the time interval $I = (0, 1)$ and consider a multiple shooting time domain decomposition $0 = \tau_0 < \tau_1 < \tau_2 = 1$ of $m = 2$ multiple shooting intervals.

The constraining equation is the heat equation with nonlinear reactive term u^3:

$$((\partial_t u, \varphi)) - ((\nabla u, \nabla \varphi)) + ((u^3, \varphi)) + (u(0) - 0, \varphi(0)) = ((q, \varphi)) \quad \forall \varphi \in X.$$

We have $H = L^2(\Omega)$, $V = H_0^1(\Omega)$, $R = L^2(\Omega)$, and $Q = L^2(I, L^2(\Omega))$. In the framework of multiple shooting for this problem we denote $\tilde{s} := (s^0, s^1)$ and $\tilde{q} := (q^0, q^1)$, and the reformulated cost functional reads

$$\bar{J}(\tilde{s}, \tilde{q}) := \sum_{j=0}^{1} \frac{1}{2} \int_{I_j} \|S_j(s^j, q^j)(t) - \bar{u}(t)\|^2 dt + \sum_{j=0}^{1} \frac{1}{2} \int_{I_j} \|q^j(t)\|^2 dt.$$

The intervalwise state equations read for $j = 0, 1$

$$((\partial_t u^j, \varphi))_j - ((\nabla u^j, \nabla \varphi))_j + (((u^j)^3, \varphi))_j + (u^j(\tau_j) - s^j, \varphi(0)) = ((q^j, \varphi))_j \quad \forall \varphi \in X^j. \tag{4.36}$$

The matching conditions are given by

$$\begin{aligned} (s^0 - 0, \varphi_0) &= 0 \qquad \forall \varphi_0 \in H, \\ (s^1 - \bar{S}_0(s^0, q^0), \varphi_1) &= 0 \qquad \forall \varphi_1 \in H. \end{aligned}$$

Now we have everything at hand to set up the Lagrangian of the problem with $\tilde{p} = (p^0, p^1)$, $p^0, p^1 \in H$:

$$\begin{aligned} \mathcal{L}(\tilde{q}, \tilde{s}, \tilde{p}) := & \sum_{j=0}^{1} \frac{1}{2} \int_{I_j} \|S_j(s^j, q_j)(t) - \bar{u}(t)\|_H^2 dt \\ & + \sum_{j=0}^{1} \frac{1}{2} \int_{I_j} \|q^j(t)\|^2 dt - \left\{ (s^0 - 0, p^0) + (s^1 - \bar{S}_0(s^0, q^0), p^1) \right\}. \end{aligned}$$

The corresponding optimality system reads

$$\begin{aligned}
(s^0, \varphi) &= 0 && \forall \varphi \in H,\\
(s^1 - \bar{S}_0(s^0, q^0), \varphi) &= 0 && \forall \varphi \in H,\\
((S'_{0s^0}(\psi), S_0(s^0, q^0) - \bar{u}))_0 - (\psi, p^0) + (\bar{S}'_{0s^0}(\psi), p^1) &= 0 && \forall \psi \in H,\\
((S'_{1s^1}(\psi), S_1(s^1, q^1) - \bar{u}))_1 - (\psi, p^1) &= 0 && \forall \psi \in H,\\
((S'_{0q^0}(\chi), S_0(s^0, q^0) - \bar{u}))_0 + ((q^0, \chi))_0 + (\bar{S}'_{0q^0}(\chi), p^1) &= 0 && \forall \chi \in Q^0,\\
((S'_{1q^1}(\chi), S_1(s^1, q^1) - \bar{u}))_1 + ((q^1, \chi))_1 &= 0 && \forall \chi \in Q^1.
\end{aligned}$$

We can easily verify by differentiation of the intervalwise state equation (4.36) that the directional derivatives of the solution operators are determined by the following equations. For given $j = 0, 1$, $s^j \in H$ and $q^j \in Q^j$ – and thus given $u^j \in X^j$ – and given directions $\varphi \in H$ and $\psi \in Q^j$ the following identities hold:

$$\begin{aligned}
&((\partial_t S'_{js^j}(\varphi), v))_j + ((\nabla S'_{js^j}(\varphi), \nabla v))_j + ((3(u^j)^2 S'_{js^j}(\varphi), v))_j\\
&\qquad + (S'_{js^j}(\varphi)(\tau_j) - \varphi, v(\tau_j)) = 0 \quad \forall v \in X^j,
\end{aligned}$$

$$\begin{aligned}
&((\partial_t S'_{jq^j}(\psi), v))_j + ((\nabla S'_{jq^j}(\psi), \nabla v))_j + ((3(u^j)^2 S'_{jq^j}(\psi), v))_j\\
&\qquad + (S'_{jq^j}(\psi)(\tau_j), v(\tau_j)) = ((\psi, v))_j \quad \forall v \in X^j.
\end{aligned}$$

These equations correspond to the abstract interpretation of the operators in Lemma 4.3. Now, the introduction of the adjoint operators allows us to reformulate the optimality system as follows:

$$\begin{aligned}
(s^0, \varphi) &= 0 && \forall \varphi \in H,\\
(s^1 - \bar{S}_0(s^0, q^0), \varphi) &= 0 && \forall \varphi \in H,\\
(S'^*_{0s^0}(S_0(s^0, q^0) - \bar{u}), \psi) - (p^0, \psi) + (\bar{S}'^*_{0s^0}(p^1), \psi) &= 0 && \forall \psi \in H,\\
(S'^*_{1s^1}(S_1(s^1, q^1) - \bar{u}), \psi) - (p^1, \psi) &= 0 && \forall \psi \in H,\\
\langle S'^*_{0q^0}(S_0(s^0, q^0) - \bar{u}), \chi \rangle_{Q^{0*} \times Q^0} + ((q^0, \chi))_0 + (\bar{S}'^*_{0q^0}(p^1), \chi) &= 0 && \forall \chi \in Q^0,\\
\langle S'^*_{1q^1}(S_1(s^1, q^1) - \bar{u}), \chi \rangle_{Q^{1*} \times Q^1} + ((q^1, \chi))_1 &= 0 && \forall \chi \in Q^1.
\end{aligned} \tag{4.37}$$

For the sake of completeness, let us shortly present the equations that allow us to calculate the evaluations of the adjoint operators. Let for given $j = 0, 1$, $u^j \in X^j$ and $r \in X^{j*}$, $\xi \in H$ the functions $x \in X^j$ and $y \in X^j$ solve the following equations for all $\varphi \in X^j$:

$$\begin{aligned}
-((\partial_t x, \varphi))_j + ((\nabla x, \nabla \varphi))_j + ((3(u^j)^2 x, \varphi))_j + (x(\tau_{j+1}), \varphi(\tau_{j+1})) &= \langle r, \varphi \rangle_{X^{j*} \times X^j},\\
-((\partial_t y, \varphi))_j + ((\nabla y, \nabla \varphi))_j + ((3(u^j)^2 y, \varphi))_j + (y(\tau_{j+1}) - \xi, \varphi(\tau_{j+1})) &= 0.
\end{aligned}$$

For the adjoint operators the following equations hold:

$$\begin{aligned}
(S'^*_{js^j}(r), \rho) &= (x(\tau_j), \rho) && \forall \rho \in H,\\
\langle S'^*_{jq^j}(r), \chi \rangle_{Q^{j*} \times Q^j} &= (x, \chi)_{Q^j} && \forall \chi \in Q^j,\\
(\bar{S}'^*_{js^j}(\xi), \rho) &= (y(\tau_j), \rho) && \forall \rho \in H,\\
\langle \bar{S}'^*_{jq^j}(\xi), \chi \rangle_{Q^{j*} \times Q^j} &= (y, \chi)_{Q^j} && \forall \chi \in Q^j.
\end{aligned}$$

Application of Theorem 4.5 finally yields the simple direct multiple shooting formulation

$$\begin{aligned}
(u^0(\tau_0) - s^0, \varphi) &= 0 \quad \forall \varphi \in H, \\
(u^0(\tau_1) - s^1, \varphi) &= 0 \quad \forall \varphi \in H, \\
(u^1(\tau_2) - s^2, \varphi) &= 0 \quad \forall \varphi \in H, \\
(z^0(\tau_0) - p^0, \psi) &= 0 \quad \forall \psi \in H, \\
(z^1(\tau_1) - p^1, \psi) &= 0 \quad \forall \psi \in H, \\
(q^0, \chi)_{Q^0} + (z^0, \chi) &= 0 \quad \forall \chi \in Q^0, \\
(q^1, \chi)_{Q^1} + (z^1, \chi) &= 0 \quad \forall \chi \in Q^1,
\end{aligned}$$

where the dual states z^0 and z^1 and are obtained from the following equations: For all $\varphi \in X^0$ resp. X^1:

$$\begin{aligned}
-((\partial_t z^0, \varphi))_0 + ((\nabla z^0, \nabla \varphi))_0 + ((3(u^0)^2 z^0, \varphi))_0 + (z^0(\tau_1) - p_1, \varphi(\tau_1)) &= ((u^0 - \bar{u}, \varphi))_0, \\
-((\partial_t z^1, \varphi))_1 + ((\nabla z^1, \nabla \varphi))_1 + ((3(u^1)^2 z^1, \varphi))_1 + (z^1(\tau_2) - p_2, \varphi(\tau_2)) &= ((u^1 - \bar{u}, \varphi))_1.
\end{aligned}$$

So far, we have developed two different multiple shooting approaches for PDE constrained optimization problems in function space. We have discussed the similarities and differences of direct and indirect multiple shooting in the previous section. For the numerical solution, time and space have to be discretized appropriately as it is discussed in the next chapter.

5 Space-Time Finite Element Discretization

This chapter is devoted to the discretization of states and controls in time and space. To be precise, we discuss Galerkin finite element discretizations which are crucial for the development of the a posteriori error estimator later on.

In the first Section 5.1 we explain the discontinuous Galerkin method of degree r (dG(r) method) for the temporal discretization of the states of the intervalwise problems (4.8). Thereafter, Section 5.2 points out the space discretization of the semidiscrete problems obtained by the previously mentioned time discretization. It is followed by Section 5.3 which deals on the one hand with the discretization of the multiple shooting variables in Subsection 5.3.1 and on the other hand with the time-space discretization of the states. In this context, we consider dynamically changing meshes (cf. Subsection 5.3.2) and interval wise constant meshes (cf. Subsection 5.3.3). Furthermore, Section 5.4 deals shortly and rather abstract with the discretization of the control space Q. Finally, Section 5.5 specifies a concrete time stepping scheme which was used for the implementation.

Remark 5.1. (Discretization for direct and indirect multiple shooting) From the previous chapter, and especially Figure 4.1, we have seen how the equations of direct and indirect approach are closely related. In fact, the equations of the optimality system (4.8) of the indirect approach cover the equations of the direct multiple shooting approach. Therefore, we consider only the discretization of the indirect multiple shooting approach in the following presentation.

Remark 5.2. (Commutability of discretization and optimization) Galerkin discretization schemes have the pleasant property that discretization and dualization interchange. Therefore, we are allowed to consider the discretization of the optimality system (4.8) directly instead of discretizing the optimal control problem (4.2) first and calculating the discrete optimality system afterwards. In other words, for Galerkin discretizations the *first-optimize-then-discretize* approach is equivalent to the *first-discretize-then-optimize* approach.

5.1 Time Discretization

In this section we explain the *discontinuous Galerkin method of degree* r (dG(r) method) which is defined by the use of discontinuous test and trial functions of degree r. A detailed derivation and description of discontinuous time discretization methods can be found in the textbook of Eriksson, Estep, Hansbo, and Johnson [17]. For the following description of the dG(r) method, we need some preliminaries and notational framework. First, let us chose a partition of the closure of the time interval $\bar{I}_j = [\tau_j, \tau_{j+1}]$:

$$\bar{I}_j = \{\, \tau_j \,\} \cup I_{j,1} \cup I_{j,2} \cdots \cup I_{j,n_j-1} \cup I_{j,n_j},$$

with $I_{j,l} = (t_{j,l-1}, t_{j,l}]$,

$$\tau_j = t_{j,0} < t_{j,1} < \cdots < t_{j,n_j-1} < t_{j,n_j} = \tau_{j+1}$$

and $k_{j,l} := |I_{j,l}|$. The discretization parameter is denoted by k, and is a piecewise constant function defined through the length of the subintervals $k\Big|_{I_{j,l}} := k_{j,l}$.

With the subintervals at hand, let us define the following spaces, which we need for the definition of the trial and test functions in the following sections. Let $\mathcal{P}_r(I_{j,l}, V)$ denote the space of polynomials on $I_{j,l}$ up to order r with values in V. We define the spaces of piecewise polynomials as follows:

$$\tilde{X}_{jk}^r := \left\{ v_k^j \in L^2(I_j, H) \;\middle|\; v_k^j\Big|_{I_{j,l}} \in \mathcal{P}_r(I_{j,l}, V),\, l = 1, \ldots n_j \text{ and } v_k^j(\tau_j) \in H \right\}.$$

Remark 5.3. We should remark that $\tilde{X}_k^{j,r} \not\subset X^j$, because the piecewise defined functions of this space are not necessarily continuous.

The discontinuous Galerkin method of degree r calculates a piecewise polynomial solution in the space $\tilde{X}_k^{j,r}$. The discontinuities of the functions $v_k^j \in \tilde{X}_k^{j,r}$ on the interior time nodes of I_j are considered by means of the following notation:

$$v_{k,l}^{j+} := \lim_{t \searrow 0} v_k^j(t_{j,l} + t), \quad v_{k,l}^{j-} := \lim_{t \nearrow 0} v_k^j(t_{j,l} + t) = v_k^j(t_{j,l}), \quad [v_k^j]_l := v_{k,l}^{j+} - v_{k,l}^{j-}.$$

For example for the dG(0) method this notation can be interpreted as follows: $v_{k,l}^{j+}$ is the limit "from above", $v_{k,l}^{j-}$ is the limit "from below" and $[v_k^j]_l$ is the "jump" in $v_k^j(t)$ at the node $t_{j,l}$. We illustrate this in Figure 5.1. We can now formulate the dG(r) method for the

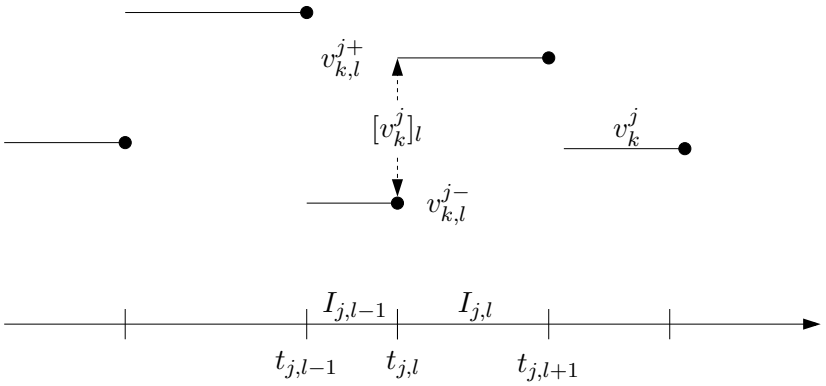


Figure 5.1: Notation for the dG(0) method.

discretization of the intervalwise state equations (4.16) as follows: Find for given control $q_k^j \in Q^j$ a state $u_k^j \in \tilde{X}_k^{j,r}$ such that

$$\sum_{l=1}^{n_j} (\!(\partial_t u_k^j, \varphi_k)\!)_{j,l} + a(u_k^j)(\varphi_k) + b(q_k^j)(\varphi_k) + \sum_{l=0}^{n_j-1} ([u_k^j]_l, \varphi_{k,l}^+) + (u_{k,0}^{j-}, \varphi_{k,0}^-)$$
$$= (\!(f, \varphi_k)\!)_j + (s_j, \varphi_{k,0}^-) \quad \forall \varphi_k \in \tilde{X}_k^{j,r}, \quad (5.1)$$

where $((v,w))_{j,l} := \int_{I_{j,l}} (v(t), w(t))dt$.

Existence and uniqueness of the time discrete solution $u_k^j \in \tilde{X}_k^{j,r}$ of (5.1) is proved in the book of Thomee [40]. In this book, uniqueness is shown by standard arguments, and the existence of a solution is concluded by reducing (5.1) to a finite dimensional problem by application of an eigenspace decomposition of the operator A as defined in (2.2).

Remark 5.4. The dG(r) approach as stated in (5.1) incorporates the initial value into the definition of $\tilde{X}_k^{j,r}$. A more common, equivalent description, like in [40] and [17], eliminates the initial value from the definition of $\tilde{X}_k^{j,r}$:

$$\sum_{l=1}^{n_j} ((\partial_t u_k^j, \varphi_k))_{j,l} + a(u_k^j)(\varphi_k) + b(q_k^j)(\varphi_k) + \sum_{l=1}^{n_j-1} ([u_k^j]_l, \varphi_{k,l}^+) + (u_{k,0}^{j+}, \varphi_{k,0}^+) \\ = ((f, \varphi_k))_j + (s_j, \varphi_{k,0}^+) \quad \forall \varphi_k \in \tilde{X}_k^{j,r}.$$

The equivalence of both schemes can be shown by elementary calculus.

Consideration of the time discretization of the restricted optimality system (4.8) yields the following semi discrete equations:

Primal equation:

$$\sum_{l=1}^{n_j} ((\partial_t u_k^j, \varphi_k))_{j,l} + a(u_k^j)(\varphi_k) + b(q_k^j)(\varphi_k) + \sum_{l=0}^{n_j-1} ([u_k^j]_l, \varphi_{k,l}^+) + (u_{k,0}^{j-}, \varphi_{k,0}^-) \\ = ((f, \varphi_k))_j + (s_j, \varphi_{k,0}^-) \quad \forall \varphi_k \in \tilde{X}_k^{j,r}. \quad (5.2a)$$

Dual equation:

$$\sum_{l=1}^{n_j} -((\partial_t z_k^j, \psi_k))_{j,l} + a_u'(u_k^j)(\psi_k, z_k^j) - \sum_{l=0}^{n_j-1} ([z_k^j]_l, \psi_{k,l}^-) + (z_{k,n_j}^{j-}, \psi_{k,n_j}^-) \\ = \alpha_1 J_1^{j\prime}(u_k^j)(\psi_k) + (\lambda_{j+1}, \psi_{k,n_j}^-) \quad \forall \psi_k \in \tilde{X}_k^{j,r}. \quad (5.2b)$$

Control equation:

$$b_q'(q_k^j)(\chi_k, z_k^j) - \alpha_3 (q_k^j, \chi_k)_{Q^j} = 0 \quad \forall \chi_k \in Q^j. \quad (5.2c)$$

Remark 5.5. This rather abstract discrete formulation for the general case of the dG(r) discretization method is specified in Section 5.5 for $r = 0$ which is the implicit Euler time stepping scheme.

Remark 5.6. (Evaluation of the integrals) The evaluation of the integrals of primal and dual equations is done by application of a quadrature formula, for example the box rule.

Remark 5.7. (Semi discrete formulation of (4.5)) For purpose of later use, let us shortly discuss, how the semi discretization of the non restricted problem on the whole time interval can be formulated in terms of the above introduced notation. Let us denote the intervalwise restriction of the states and controls as before, and let us accordingly denote the time discrete space on the time interval I by

$$\tilde{X}_k^r := \left\{ v_k \in L^2(I,H) \;\middle|\; v_k\big|_{I_{j,l}} \in \mathcal{P}_r(I_{j,l}, V),\ j = 0, \dots m-1,\ l = 1, \dots n_j, \text{ and } v_k(0) \in H \right\}.$$

System (4.5) reads in time discrete formulation as follows:

Primal equation:

$$\sum_{j=0}^{m-1}\left\{\sum_{l=1}^{n_j}(\!(\partial_t u_k^j,\varphi_k^j)\!)_{j,l}+a(u_k^j)(\varphi_k^j)+b(q_k^j)(\varphi_k^j)+\sum_{l=1}^{n_j-1}([u_k^j]_l,\varphi_{k,l}^{j+})\right\}$$
$$+([u_k^0]_0,\varphi_{k,0}^{0+})+\sum_{j=0}^{m-2}(u_{k,0}^{(j+1)+}-u_{k,n_j}^{j-},\varphi_{k,0}^{(j+1)+})+(u_{k,0}^{0-},\varphi_{k,0}^{0-})$$
$$=(\!(f,\varphi_k^j)\!)+(u_0,\varphi_{k,0}^{0-})\quad\forall\varphi_k^j\in\tilde{X}_k^r.\quad(5.3a)$$

Dual equation:

$$\sum_{j=0}^{m-1}\left\{\sum_{l=1}^{n_j}-(\!(\partial_t z_k^j,\psi_k^j)\!)_{j,l}+a_u'(u_k^j)(\psi_k^j,z_k^j)-\sum_{l=1}^{n_j-1}([z_k^j]_l,\psi_{k,l}^{j-})\right\}$$
$$-([z_k^0]_0,\psi_{k,0}^{0-})-\sum_{j=0}^{m-2}(z_{k,0}^{(j+1)+}-z_{k,n_j}^{j-},\psi_{k,0}^{j-})+(z_{k,n_{m-1}}^{(m-1)-},\psi_{k,n_{m-1}}^{(m-1)-})$$
$$=\sum_{j=0}^{m-1}\alpha_1 J_1^{j\prime}(u_k^j)(\psi_k^j)+\alpha_2 J_2'(u^{m-1}(T)(\psi_{k,n_{m-1}}^{(m-1)-})\quad\forall\psi_k\in\tilde{X}_k^r.\quad(5.3b)$$

Control equation:

$$b_q^{'}(q_k)(\chi_k,z_k)-(q_k,\chi_k)_Q=0\quad\forall\chi_k\in Q.\quad(5.3c)$$

In the next section we develop the spatial discretization of the continuous spaces V and H which still remain in the definition of the semidiscrete spaces $\tilde{X}_k^{j,r}$.

5.2 Space Discretization

The spatial discretization of the Hilbert space V is done by defining finite dimensional subspaces $V_h^s\subseteq V$ consisting of *finite elements of maximal order s*. The spatial discretizations may differ for each interval $I_{j,l}$, which we take into account by adding additional indices j and l in the notation of the discrete space whenever necessary. For now, we merely consider an arbitrary discrete space $V_h^s\subseteq V$ for a given triangulation.

In this section we introduce the triangulation of the bounded spatial domain $\Omega\subset\mathbb{R}^d$, $d=2,3$, and develop appropriate finite element spaces on this triangulation. We assume the boundary of the domain, $\partial\Omega$, to be of polygonal shape. The general case is not considered in this thesis but is for example derived in the book of Braess [12]. Depending on the dimension of the space, we decompose the domain Ω into quadrilaterals (for $d=2$) or hexalaterals (for $d=3$) denoted by K which cover the whole domain Ω. The corresponding triangulation is labeled $\mathcal{T}_h=\{K\}$, where the parameter h is a cellwise constant function and gives information on the diameter of the current cell, $h\Big|_K=h_K:=\operatorname{diam}K$.

A mathematical formulation of this context is given in [12] in terms of the following definition:

Definition 5.1. (Regular triangulation) A triangulation $\mathcal{T}_h = \{K\}$ of Ω is called regular if the following properties hold:

1. $\bar{\Omega} = \bigcup_{K \in \mathcal{T}_h}$.
2. If for two cells K_1 and K_2, $K_1 \neq K_2$, the intersection $K_1 \cap K_2 = p$, $p \in \Omega$, then p is a vertice of both cells K_1 and K_2.
3. If for two cells K_1 and K_2, $K_1 \neq K_2$, the intersection $K_1 \cap K_2$ consists of more than one point, then $K_1 \cap K_2$ is a face of both cells K_1 and K_2.

Remark 5.8. The parameter h is often considered as the maximum diameter of all cells in $\mathcal{T}_h$, that is $h := \max_{K \in \mathcal{T}_h} \operatorname{diam} K$.

We are especially interested in families of triangulations arising from the successive refinement of cells. Therefore, we request the following properties for the triangulations:

Definition 5.2. (Quasi uniform family of triangulations) A family of regular triangulations $\{\mathcal{T}_h\}$ with $h \searrow 0$ is called quasi uniform, if there exist $\kappa > 0$ such that each $K \in \bigcup_h \mathcal{T}_h$ contains a circle of radius ρ_K with

$$\rho_K \geq \frac{h_K}{\kappa}.$$

Definition 5.3. (Uniform family of triangulations) A family of triangulations $\{\mathcal{T}_h\}$ with $h \searrow 0$ is called uniform, if there exist $\kappa > 0$ such that each $K \in \bigcup_h \mathcal{T}_h$ contains a circle of radius ρ_K with

$$\rho_K \geq \frac{h}{\kappa},$$

where h has to be interpreted in the sense of Remark 5.8.

Remark 5.9. The definitions of quasi uniformity and uniformity above are merely used for theoretical purpose when considering approximation properties of the finite element spaces and performing a priori error analysis. We do not persecute this theoretical investigations, but refer to the literature mentioned before. Nevertheless, our triangulations fulfill quasi uniformity in practice.

For the purpose of local refinement we have to weaken the last condition in Definition 5.1. We allow *hanging nodes*, that are nodes that lie in the middle of a face of a cell K, and assume furthermore that the triangulation $\mathcal{T}_h$ is obtained from the triangulation $\mathcal{T}_{2h}$ by *patchwise refinement*. A patch denotes a set of four cells (for $d = 2$) or eight cells (for $d = 3$) obtained by refinement of a coarser cell. This patchwise organization is of importance when we consider patchwise interpolation for higher order approximation of the exact solution. This is needed for the evaluation of a posteriori error estimators in Chapter 7. An example for a triangulation with patchwise organization of the cells is given in Figure 5.2.

In the context of finite elements, we consider the cells of a triangulation as transformations of a reference cell $\hat{K}$, for example the unit square in the case of quadrilaterals. We denote

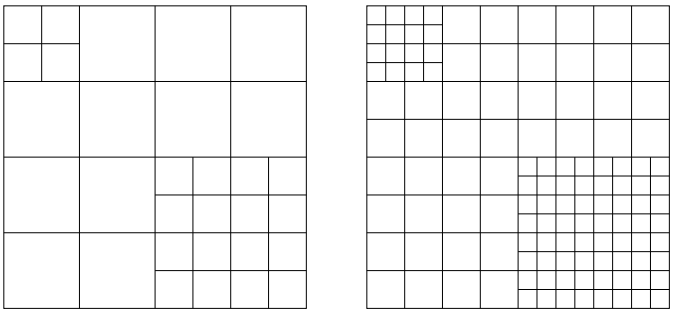


Figure 5.2: Triangulation with patchwise cell organization (right) obtained from a coarser triangulation (left) by global refinement.

the transformation that maps the reference cell onto the current cell K by $\varphi_K : \hat{K} \to K$, see Figure 5.3, and define the polynomial space on the reference cell $\hat{K}$ by

$$\hat{\mathcal{Q}}_s(\hat{K}) := \operatorname{span}\left\{ \prod_{k=1}^{d} x_k^{\alpha_k} \;\middle|\; \alpha_k \in \{0, 1, \ldots, s\} \right\}.$$

Then the corresponding space of polynomials on K is defined by use of the transformation φ:

$$\mathcal{Q}_s(K) := \left\{ v \;\middle|\; v \circ \varphi \in \hat{\mathcal{Q}}_s(\hat{K}) \right\}.$$

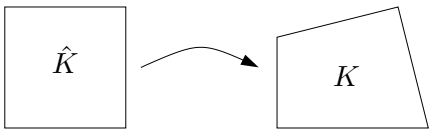


Figure 5.3: Transformation of the reference cell into an arbitrary cell.

The finite element space consists of continuous piecewise polynomial functions of order up to s, that is

$$V_h^s := \left\{ v_h \in V \;\middle|\; v_h\big|_K \in \mathcal{Q}_s(K), K \in \mathcal{T}_h \right\}.$$

Remark 5.10. The admission of hanging nodes leads to problems with respect to the continuity on the faces. In order to make a finite element function globally continuous, we have to make sure that the hanging nodes have values that are compatible with the adjacent nodes on the vertices, such that the function has no jump when coming from the refined cells to the adjacent coarse cell. Therefore, we eliminate the degrees of freedom corresponding to the hanging nodes and determine the values at the hanging nodes from the adjacent degrees of freedom by an interpolation procedure after the solution process.

5.3 Discretization of Time and Space

In this section we finally merge the discretization schemes for time and space from Sections 5.1 and 5.2 in order to obtain a time-space discrete formulation of problem (4.8) with fully discretized states. We present two different discretization schemes, one of which has been proven to be more efficient with respect to the implementation and the computational effort. First, we consider the discretization of the multiple shooting variables. This can be done independently from the discretization of the states on the intervals and offers a large variety of possibilities for the choice of the mesh. Second, we present the discretization with dynamically changing meshes in every time step which are only related by a common coarse grid. And finally, a special case of the dynamically changing meshes, namely the case of intervalwise constant meshes, is derived.

5.3.1 Discretization of the Multiple Shooting Variables

The intervalwise consideration of the time space discrete intervalwise problems possibly yields different adjacent spatial meshes on the multiple shooting nodes. This is due to the fact, that the local mesh size of $\mathcal{T}_h^{j,l}$ is determined by refinement indicators on the interval I_j. Therefore, the final mesh of I_j and the first mesh of I_{j+1} are in general different and only related by a common coarse grid. As a consequence, we have a certain freedom to choose the discretization for the shooting variables connecting the two. In particular, we end up with different discretizations of H in each shooting node τ_j. We denote these spaces by H_h^j and the corresponding meshes by $\mathcal{T}_h^{\tau_j}$. The spatial discretization is, as before, performed with a continuous Galerkin approach of degree s. The spatial discrete multiple shooting variables are denoted by $s_h^j, \lambda_h^j \in H_h^j$.

Remark 5.11. (Choice of the node meshes) While the mesh used for H_h^j can be chosen independent of the neighboring intervals, we use the trace mesh of either the left interval or the right for practical purposes in the section on numerical experiments.

5.3.2 Dynamically Changing Spatial Meshes

A spatial discretization with dynamically changing meshes in time suggests itself. The triangulation corresponding to the timepoint $t_{j,l}$ is denoted by $\mathcal{T}_h^{j,l}$ and the appropriate finite element space by $V_h^{j,l,s}$ as illustrated in Figure 5.4.

With this notation at hand, we are able to define the following fully discrete spaces:

$$\tilde{X}_{kh}^{j,r,s} := \left\{ v_{kh} \in L^2(I_j, H) \;\middle|\; v_{kh}\Big|_{I_{j,l}} \in \mathcal{P}_r(I_{j,l}, V_h^{j,l,s}),\ l = 1, \dots n_j, \text{ and } v_{kh}(\tau_j) \in V_h^{j,0,s} \right\}.$$

It can directly be seen that due to the conformity $V_h^{j,l,s} \subseteq V$ the inclusion $\tilde{X}_{kh}^{j,r,s} \subseteq \tilde{X}_k^{j,r}$ holds for the fully discrete space. For discretizations of this type, the notation cG(s)dG(r) has been introduced in [17] because of the cG(s) discretization in space and the dG(r) discretization in time. The cG(s)dG(r) formulation of problem (4.16) is consequently obtained from the

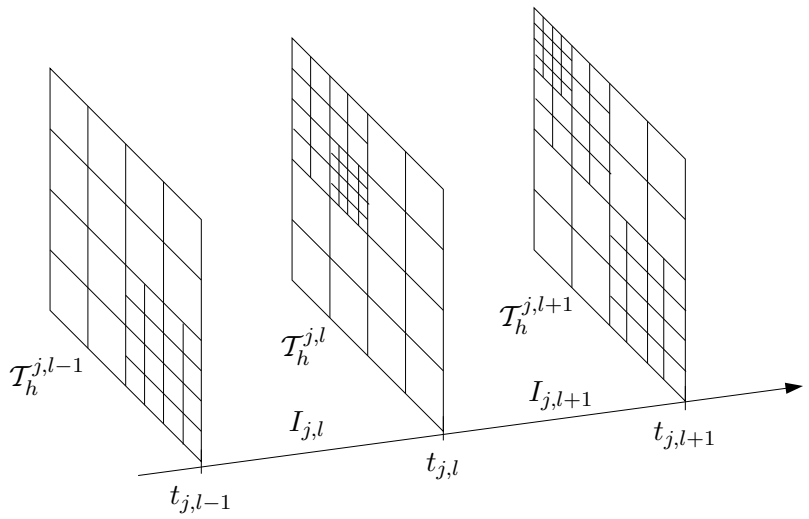


Figure 5.4: Dynamically changing spatial meshes in time.

semidiscrete formulation (5.1) by replacing the space $\tilde{X}_k^{j,r,s}$ by its fully discrete equivalent $\tilde{X}_{kh}^{j,r,s}$ and adding an index h to the variables and test functions:

Find for given control $q_{kh}^j \in Q^j$ a state $u_{kh}^j \in \tilde{X}_{kh}^{j,r,s}$ such that

$$\sum_{l=1}^{n_j} (\!(\partial_t u_{kh}^j, \varphi_{kh})\!)_{j,l} + a(u_{kh}^j)(\varphi_{kh}) + b(q_{kh}^j)(\varphi_{kh}) + \sum_{l=0}^{n_j-1} ([u_{kh}^j]_l, \varphi_{kh,l}^+) + (u_{kh,0}^{j-}, \varphi_{kh,0}^-) = (\!(f, \varphi_{kh})\!)_j + (s_h^j, \varphi_{kh,0}^-) \quad \forall \varphi_{kh} \in \tilde{X}_{kh}^{j,r,s}.$$

By the same procedure, we obtain the cG(s)dG(r) formulation from the semidiscrete optimality system (5.2):

Primal equation:

$$\sum_{l=1}^{n_j} (\!(\partial_t u_{kh}^j, \varphi_{kh})\!)_{j,l} + a(u_{kh}^j)(\varphi_{kh}) + b(q_{kh}^j)(\varphi_{kh}) + \sum_{l=0}^{n_j-1} ([u_{kh}^j]_l, \varphi_{kh,l}^+) + (u_{kh,0}^{j-}, \varphi_{kh,0}^-) = (\!(f, \varphi_{kh})\!)_j + (s_h^j, \varphi_{kh,0}^-) \quad \forall \varphi_{kh} \in \tilde{X}_{kh}^{j,r,s}. \tag{5.4a}$$

Dual equation:

$$\sum_{l=1}^{n_j} -(\!(\partial_t z_{kh}^j, \psi_{kh})\!)_{j,l} + a_u'(u_{kh}^j)(\psi_{kh}, z_{kh}^j) - \sum_{l=0}^{n_j-1} ([z_{kh}^j]_l, \psi_{kh,l}^-) + (z_{kh,n_j}^{j-}, \psi_{kh,n_j}^-) = \sum_{j=0}^{m-1} \alpha_1 J_1^{j\prime}(u_{kh}^j)(\psi_{kh}) + (\lambda_h^{j+1}, \psi_{kh,n_j}^-) \quad \forall \psi_{kh} \in \tilde{X}_{kh}^{j,r,s}. \tag{5.4b}$$

Control equation:

$$b_q'(q_{kh}^j)(\chi_{kh}, z_{kh}^j) - \alpha_3 (q_{kh}^j, \chi_{kh})_{Q^j} = 0 \quad \forall \chi_{kh} \in Q^j. \tag{5.4c}$$

Remark 5.12. (Time-space discrete formulation of (4.5)) To complete this presentation and for purpose of later usage, the time space discrete formulation of problem (4.5) is gained by the same procedure applied to the semi discrete system (5.3). We introduce the fully discrete space on the time interval I by

$$\tilde{X}_{kh}^{r,s} := \left\{ v_k \in L^2(I,H) \;\middle|\; v_k\big|_{I_{j,l}} \in \mathcal{P}_r(I_{j,l}, V_h^{j,l,s}),\ j=0,\dots m-1, l=1,\dots n_j,\ v_k(0) \in V_h^{j,0,s} \right\}$$

and write the optimality system in time-space discrete formulation as follows:

Primal equation:

$$\begin{aligned}\sum_{j=0}^{m-1} \left\{ \sum_{l=1}^{n_j} (\!(\partial_t u_{kh}^j, \varphi_{kh}^j)\!)_{j,l} + a(u_{kh}^j)(\varphi_{kh}^j) + b(q_{kh}^j)(\varphi_{kh}^j) + \sum_{l=1}^{n_j-1} ([u_{kh}^j]_l, \varphi_{kh,l}^{j+}) \right\} \\ + ([u_{kh}^0]_0, \varphi_{kh,0}^{0+}) + \sum_{j=0}^{m-2} (u_{kh,0}^{(j+1)+} - u_{kh,n_j}^{j-}, \varphi_{kh,0}^{(j+1)+}) + (u_{kh,0}^{0-}, \varphi_{kh,0}^{0-}) \\ = (\!(f, \varphi_{kh}^j)\!) + (u_0, \varphi_{kh,0}^{0-}) \quad \forall \varphi_{kh}^j \in \tilde{X}_{kh}^{r,s}. \quad (5.5a)\end{aligned}$$

Dual equation:

$$\begin{aligned}\sum_{j=0}^{m-1} \left\{ \sum_{l=1}^{n_j} -(\!(\partial_t z_{kh}^j, \psi_{kh}^j)\!)_{j,l} + a_u'(u_{kh}^j)(\psi_{kh}^j, z_{kh}^j) - \sum_{l=1}^{n_j-1} ([z_{kh}^j]_l, \psi_{kh,l}^{j-}) \right\} \\ - ([z_{kh}^0]_0, \psi_{kh,0}^{0-}) - \sum_{j=0}^{m-2} (z_{kh,0}^{(j+1)+} - z_{kh,n_j}^{j-}, \psi_{kh,0}^{j-}) + (z_{kh,n_{m-1}}^{(m-1)-}, \psi_{kh,n_{m-1}}^{(m-1)-}) \\ = \sum_{j=0}^{m-1} \alpha_1 J_1^{j\prime}(u_{kh}^j)(\psi_{kh}^j) + \alpha_2 J_2'(u^{m-1}(T)(\psi_{kh,n_{m-1}}^{(m-1)-}) \quad \forall \psi_{kh} \in \tilde{X}_{kh}^{r,s}. \quad (5.5b)\end{aligned}$$

Control equation:

$$b_q^{'}(q_{kh})(\chi_{kh}, z_{kh}) - (q_{kh}, \chi_{kh})_Q = 0 \quad \forall \chi_{kh} \in Q. \quad (5.5c)$$

Remark 5.13. (Discrete matching conditions) The discrete matching conditions are finally, under consideration of Remark 5.11, given as

$$\begin{aligned}(s_h^0 - u_0, v_h) &= 0 \quad \forall v_h \in H_h^0, \\ (s_h^{j+1} - u_{kh,n_j}^{j-}(\tau_{j+1}), v_h) &= 0 \quad \forall v_h \in H_h^{(j+1)}, \quad j=0,\dots,m-1, \\ (\lambda_h^j - z_{kh,0}^{j-}(\tau_j), v_h) &= 0 \quad \forall v_h \in H_h^j, \quad j=0,\dots,m-1, \\ (\lambda_h^m, v_h) - \alpha_2 J_2^{'}(s_h^m)(v_h) &= 0 \quad \forall v_h \in H_h^m.\end{aligned}$$

Remark 5.14. (Efficiency) The application of dynamically changing meshes for multiple shooting methods is very time consuming in practice. The number of matrix assemblations is extremely large, and most of the calculation time is thus spent by calculating the system matrices and right-hand sides. A possible improvement is the consideration of intervalwise constant meshes as presented in the following subsection.

5.3.3 Intervalwise Constant Spatial Meshes

The spatial discretization with intervalwise constant meshes is a promising alternative to the choice of dynamically changing meshes. As depicted in Figure 5.5, the spatial mesh is the same for all time steps of one multiple shooting interval.

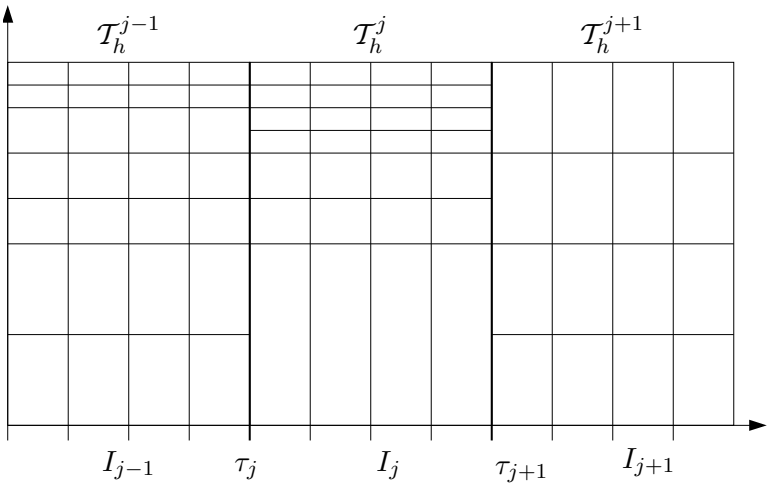


Figure 5.5: Intervalwise constant meshes in time.

On the one hand, this simplifies the notation because we no longer need the time step index l for the discretization of V, on the other hand we see later on, that this idea provides us a highly efficient framework for the implementation and solution of the discretized problems. As this approach is only a special case of the general idea of dynamically changing meshes, the formulation of the discretized intervalwise optimality systems and the matching conditions is the same as in the previous subsection.

5.4 Discretization of the Controls

In the previous sections, the control space Q^j has been kept undiscretized while the space X^j was discretized in time and space by the cG(s)dG(r) method. In the sequel, we shortly discuss possible discretizations, and the discretization method for Q^j is illustrated by means of Example 2.3. Before we proceed, we should remark that according to [22] the solution of the optimal control problem is possible without discretizing the control space, even if it is of infinite dimension.

The control spaces Q^j have been introduced by the inclusion

$$Q^j \subseteq L^2(I_j, R)$$

with a Hilbert space R. Concerning the control, we generally chose the time discretization of the control the same as for the states, that is a dG(r) discretization on the same temporal mesh. For the space discretization of R we allow a cG(p) discretization method of lower

degree than chosen for the states, that is $p \leq s$. Altogether, we chose a finite dimensional subspace $Q_d^j \subseteq Q^j$ which is defined through a cG(p)dG(r) discretization method. In the following we denote the spatial discrete equivalent of R in $t_{j,l}$ by $R_h^{j,l,p}$. In combination with the discretization of the states, we obtain a fully discrete formulation of the optimization problem.

On the basis of Example 2.3 we can see that this choice is reasonable:

Example 5.1. (Discretization of Q^j for Example 2.3) In this example the control space is given by

$$R = L^2(\Omega) \quad \text{and thus} \quad Q^j = L^2(I_j, L^2(\Omega)),$$

whereas the space for the states was defined through

$$H = L^2(\Omega) \quad \text{and} \quad V = H_0^1(\Omega).$$

From the control equation in (4.14) we retrieve the relation between the dual state z^j and the control q^j as

$$(\!(q, \chi)\!)_j = -\frac{1}{\alpha_3}(\!(z, \chi)\!)_j \quad \forall \chi \in Q^j.$$

From this context, it is consequential to chose the discretization of the control either the same as for the state z^j or a coarser one or a discretization of lower degree. The choice of a coarser triangulation in space or time is not considered in this thesis. Nevertheless, the extension of the herein mentioned multiple shooting and error estimation methods to these cases is straightforward.

Our discretizations Q_d^j of Q^j, as presented for Example 2.3, always fulfill conformity, such that the fully discrete restricted optimization problem can be derived straightforward from (5.4) by adding an additional discretization index to the states, controls and the control space. Following the notation of Meidner [29], we denote the fully discrete variables by q_{khd}^j, u_{khd}^j, z_{khd}^j and abbreviate the index khd by σ. With this notation, the fully discretized state equation reads as follows:

Find for given control $q_\sigma^j \in Q_d^j$ a state $u_\sigma^j \in \tilde{X}_{kh}^{j,r,s}$ such that

$$\sum_{l=1}^{n_j}(\!(\partial_t u_\sigma^j, \varphi_\sigma)\!)_{j,l} + a(u_\sigma^j)(\varphi_\sigma) + b(q_\sigma^j)(\varphi_\sigma) + \sum_{l=0}^{n_j-1}([u_\sigma^j]_l, \varphi_{\sigma,l}^+) + (u_{\sigma,0}^{j-}, \varphi_{\sigma,0}^-) = (\!(f, \varphi_\sigma)\!)_j + (s_h^j, \varphi_{\sigma,0}^-) \quad \forall \varphi_\sigma \in \tilde{X}_{kh}^{j,r,s},$$

and analogously we obtain the fully discrete formulation of the optimality system (5.2):

Primal equation:

$$\sum_{l=1}^{n_j}(\!(\partial_t u_\sigma^j, \varphi_\sigma)\!)_{j,l} + a(u_\sigma^j)(\varphi_\sigma) + b(q_\sigma^j)(\varphi_\sigma) + \sum_{l=0}^{n_j-1}([u_\sigma^j]_l, \varphi_{\sigma,l}^+) + (u_{\sigma,0}^{j-}, \varphi_{\sigma,0}^-) = (\!(f, \varphi_\sigma)\!)_j + (s_h^j, \varphi_{\sigma,0}^-) \quad \forall \varphi_\sigma \in \tilde{X}_{kh}^{j,r,s}. \quad (5.6a)$$

Dual equation:

$$\sum_{l=1}^{n_j} -((\partial_t z_\sigma^j, \psi_\sigma))_{j,l} + a'_u(u_\sigma^j)(\psi_\sigma, z_\sigma^j) - \sum_{l=0}^{n_j-1} ([z_\sigma^j]_l, \psi_{\sigma,l}^-) + (z_{\sigma,n_j}^{j-}, \psi_{\sigma,n_j}^-) = \alpha_1 J_1^{j\prime}(u_\sigma^j)(\psi_\sigma) + (\lambda_h^{j+1}, \psi_{\sigma,n_j}^-) \quad \forall \psi_\sigma \in \tilde{X}_{kh}^{j,r,s}. \tag{5.6b}$$

Control equation:

$$b_q'(q_\sigma^j)(\chi_\sigma, z_\sigma^j) - \alpha_3(q_\sigma^j, \chi_\sigma)_{Q^j} = 0 \quad \forall \chi_\sigma \in Q_d^j. \tag{5.6c}$$

Remark 5.15. (Fully discrete formulation of (4.5)) The fully discrete formulation of (4.5) is derived analogously from (5.5):

Primal equation:

$$\sum_{j=0}^{m-1} \left\{ \sum_{l=1}^{n_j} ((\partial_t u_\sigma^j, \varphi_\sigma^j))_{j,l} + a(u_\sigma^j)(\varphi_\sigma^j) + b(q_\sigma^j)(\varphi_\sigma^j) + \sum_{l=1}^{n_j-1} ([u_\sigma^j]_l, \varphi_{\sigma,l}^{j+}) \right\} + ([u_\sigma^0]_0, \varphi_{\sigma,0}^{0+}) + \sum_{j=0}^{m-2} (u_{\sigma,0}^{(j+1)+} - u_{\sigma,n_j}^{j-}, \varphi_{\sigma,0}^{(j+1)+}) + (u_{\sigma,0}^{0-}, \varphi_{\sigma,0}^{0-}) = ((f, \varphi_\sigma^j)) + (u_0, \varphi_{\sigma,0}^{0-}) \quad \forall \varphi_\sigma^j \in \tilde{X}_\sigma^{r,s}. \tag{5.7a}$$

Dual equation:

$$\sum_{j=0}^{m-1} \left\{ \sum_{l=1}^{n_j} -((\partial_t z_\sigma^j, \psi_\sigma^j))_{j,l} + a'_u(u_\sigma^j)(\psi_\sigma^j, z_\sigma^j) - \sum_{l=1}^{n_j-1} ([z_\sigma^j]_l, \psi_{\sigma,l}^{j-}) \right\} - ([z_\sigma^0]_0, \psi_{\sigma,0}^{0-}) - \sum_{j=0}^{m-2} (z_{\sigma,0}^{(j+1)+} - z_{\sigma,n_j}^{j-}, \psi_{\sigma,0}^{j-}) + (z_{\sigma,n_{m-1}}^{(m-1)-}, \psi_{\sigma,n_{m-1}}^{(m-1)-}) = \sum_{j=0}^{m-1} \alpha_1 J_1^{j\prime}(u_\sigma^j)(\psi_\sigma^j) + \alpha_2 J_2'(u^{m-1}(T)(\psi_{\sigma,n_{m-1}}^{(m-1)-}) \quad \forall \psi_\sigma \in \tilde{X}_\sigma^{r,s}. \tag{5.7b}$$

Control equation:

$$b_q'(q_\sigma)(\chi_\sigma, z_\sigma) - (q_\sigma, \chi_\sigma)_Q = 0 \quad \forall \chi_\sigma \in Q_d. \tag{5.7c}$$

To complete this section on the fully discrete formulation, we state the matching conditions for the fully discrete case in the following remark.

Remark 5.16. (Fully discrete formulation of the matching conditions) The discrete matching conditions are finally given as

$$\begin{aligned} (s_h^0 - u_0, v_h) &= 0 \quad \forall v_h \in H_h^0, \\ (s_h^{j+1} - u_{\sigma,n_j}^{j-}(\tau_{j+1}), v_h) &= 0 \quad \forall v_h \in H_h^{(j+1)}, \quad j = 0, \dots, m-1, \\ (\lambda_h^j - z_{\sigma,0}^{j-}(\tau_j), v_h) &= 0 \quad \forall v_h \in H_h^j, \quad j = 0, \dots, m-1, \\ (\lambda_h^m, v_h) - \alpha_2 J_2'(s_h^m)(v_h) &= 0 \quad \forall v_h \in H_h^m. \end{aligned} \tag{5.8}$$

Remark 5.17. (Notation) In view of the subsequent chapters, we introduce an additional notation for the Cartesian product of discrete spaces as given below:

$$\begin{aligned}\tilde{X}_{kh} &:= \tilde{X}_{kh}^{0,r,s} \times \cdots \times \tilde{X}_{kh}^{m-1,r,s},\\ \tilde{Q}_d &:= Q_d^0 \times \cdots \times Q_d^{m-1},\\ \tilde{H}_h &:= H_h^0 \times \cdots \times H_h^m.\end{aligned}$$

5.5 The Implicit Euler Time Stepping Scheme

In this section, we shortly describe a concrete time discretization method for $r = 0$, which we use throughout our computations. The time stepping scheme for the cG(s)dG(0) method corresponds to the *implicit Euler scheme* as presented in the sequel, where we approximate the arising integrals by the box rule. The parameter choice of $r = 0$ means to have piecewise constant functions in time such that it is straightforward to introduce the following subintervalwise notation:

$$Q_{\sigma,l}^j := q_{\sigma,l}^{j-}, \quad U_{\sigma,l}^j := u_{\sigma,l}^{j-}, \quad Z_{\sigma,l}^{j-} := z_{\sigma,l}^{j-}. \tag{5.9}$$

We remark that the functional J_1 was defined such that an interval wise splitting according to equation (2.5) is possible.

The cG(s)dG(0) scheme for problem (4.8) in fully discrete formulation is stated as follows: for all $\varphi \in V_h^{j,l,s}$, $\chi \in R_h^{j,l,p}$ let the following equations be fulfilled

Primal equation:

$l = 0$:

$$(U_0^j, \varphi) = (s_\sigma^j, \varphi),$$

$l = 1, \ldots, n_j$:

$$(U_{\sigma,l}^j, \varphi) + k_{j,l}\bar{a}(U_{\sigma,l}^j)(\varphi) + k_{j,l}\bar{b}(Q_{\sigma,l}^j)(\varphi) = (U_{\sigma,l-1}^j, \varphi) + k_{j,l}(f(t_{j,l}), \varphi),$$

Dual equation:

$l = n_j$:

$$(Z_{n_j}^j, \varphi) = (\lambda_\sigma^{j+1}, \varphi),$$

$l = 1, \ldots, n_j - 1$:

$$(Z_{\sigma,l}^j, \varphi) + k_{j,l}\bar{a}_u'(U_{\sigma,l}^j)(\varphi, Z_{\sigma,l}^j) = (Z_{\sigma,l+1}^j, \varphi) + \alpha_1 k_{j,l} F'(U_{\sigma,l}^j)(\varphi),$$

$l = 0$:

$$(Z_0^j, \varphi) = (Z_1^j, \varphi),$$

Control equation:

$$\bar{b}_q'(Q_{\sigma,l}^j)(\chi, Z_{\sigma,l}^j) = \alpha_3 (Q_{\sigma,l}^j, \chi)_R.$$

We proceed with the investigation of solution techniques for the multiple shooting problem in the next chapter.

6 Solution Techniques for the Multiple Shooting Approach

This chapter is devoted to the development and discussion of different solution techniques for the multiple shooting approach. Multiple shooting as presented in the previous chapters exploits the idea of applying standard iterative solvers to the system of equations, treating the underlying boundary or initial value problems as solved by a (not yet concretized) solution routine. In this chapter, we derive for both direct and indirect multiple shooting first the iterative solvers for the system of equations and second the solution techniques for the boundary and initial value problems.

Consequently, the first Section 6.1 deals with the solution techniques for the indirect multiple shooting approach. First of all, the solution of the discretized system of matching conditions (5.8) by Newton's method is considered in Subsection 6.1.1. In this context, the linearized system is derived, and a short pseudo matrix notation is introduced. Its solution by means of a Krylov subspace method is investigated in the subsequent Subsection 6.1.2. The necessity of a preconditioner is outlined by numerical examples, and consequently three different preconditioners are presented in the sequel. It will turn out, that most of the computational effort is spend in the solution of the intervalwise nonlinear and linear boundary value problems. Thus, different solution techniques for the intervalwise problems are discussed in the Subsections 6.1.3, 6.1.4, and 6.1.6. We investigate the idea of solving the nonlinear boundary value problems by application of Newton's method on the whole interval problem in 6.1.3 and line out the limitations of this approach in Subsection 6.1.5.

Section 6.2 is engaged with the solution techniques for the direct multiple shooting approach. We start with an analogous discussion of Newton's method for the solution of the discrete nonlinear system of matching conditions in Subsection 6.2.1 and proceed with the investigation of a preconditioned Krylov subspace method for the solution of the linearized system in Subsection 6.2.2. Furthermore, an efficient condensing technique for the linearized system is introduced in Subsection 6.2.3. This procedure allows a remarkable reduction of the solution space. The condensing technique is similar to the ODE condensing approach, and consequently we devote Subsection 6.2.4 to the restrictions which the transfer of the ODE approach to PDEs is subject to.

Finally, we close this chapter with the numerical comparison of the different solution techniques by means of numerical examples in Section 6.3.

Remark 6.1. (Notation) Throughout this chapter we consider with a few exceptions the fully discretized problem and accordingly the fully discrete matching conditions. Thus, the solution operators within have to be understood as those of the discretized problems. As

from the arguments given to the solution operators, there is no way of misunderstanding we do not denote the discrete solution operators differently from the continuous ones.

6.1 Solution Techniques for the Indirect Multiple Shooting Approach

This section deals with the development of two different solution techniques for the indirect multiple shooting approach. Both ideas start with the application of *Newton's method* to the system of matching conditions and the application of an *iterative Krylov subspace method* for the solution of the linearized system. Newton's method requires the evaluation of the residual, and the iterative linear solver needs the calculation of the directional derivatives forming the left-hand side of the linear problem. Clearly, the linear solver depends on the solution of interval wise linear boundary value problems (BVPs) while the evaluation of the residual needs the solution of nonlinear problems. The interrelation of the different subproblems is sketched in Figure 6.1.

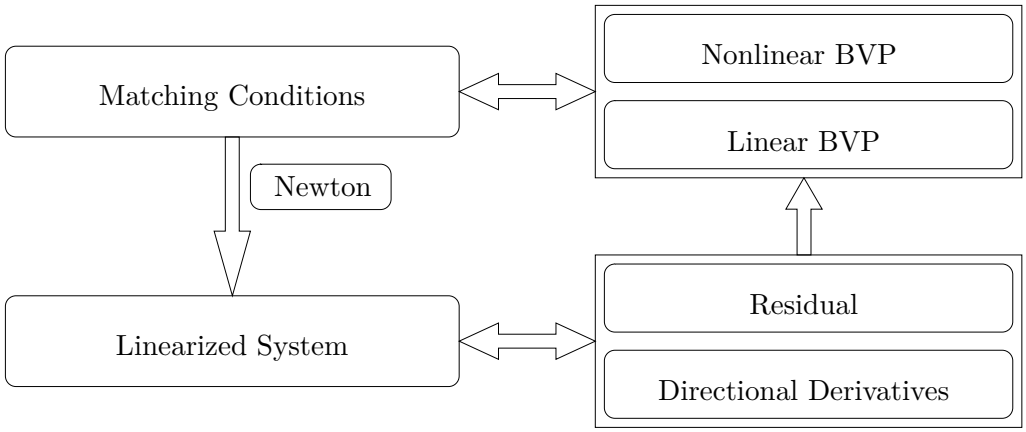


Figure 6.1: Nesting of the different solvers.

6.1.1 Solution of the Multiple Shooting System

The consideration of Newton's method as a solver for the system of matching conditions (5.8) yields the following linear system for the calculation of the Newton increment $(\Delta\tilde{s}_h, \Delta\tilde{\lambda}_h)$:

Find $(\Delta\tilde{s}_h, \Delta\tilde{\lambda}_h) \in \tilde{H}_h \times \tilde{H}_h$ such that for all $j = 0, \dots, m-1$

$$\begin{aligned}
(\Delta s_h^0, v) &= -(s_h^0 - u_0, v) && \forall v \in H_h^0,\\
(\Delta s_h^{j+1} - \bar{\Sigma}'_{u^j,s^j}(\Delta s_h^j) - \bar{\Sigma}'_{u^j,\lambda^{j+1}}(\Delta\lambda_h^{j+1}), v) &= -(s_h^{j+1} - \bar{\Sigma}_{u^j}(s_h^j, \lambda_h^{j+1}), v) && \forall v \in H_h^{j+1},\\
(\Delta\lambda_h^j - \bar{\Sigma}'_{z^j,s^j}(\Delta s_h^j) - \bar{\Sigma}'_{z^j,\lambda^{j+1}}(\Delta\lambda_h^{j+1}), v) &= -(\lambda_h^j - \bar{\Sigma}_{z^j}(s_h^j, \lambda_h^{j+1}), v) && \forall v \in H_h^j,\\
(\Delta\lambda_h^m, v) - \alpha_2 J_2''(s_h^m)(v, \Delta s_h^m) &= -(\lambda_h^m, v) - \alpha_2 J_2'(s_h^m)(v) && \forall v \in H_h^m,
\end{aligned} \tag{6.1}$$

for which we used the abbreviations

$$\begin{aligned}
\bar{\Sigma}'_{u^j,s^j}(\Delta s_h^j) &:= \bar{\Sigma}'_{u^j,s^j}(s_h^j,\lambda_h^{j+1})(\Delta s_h^j),\\
\bar{\Sigma}'_{u^j,\lambda^{j+1}}(\Delta \lambda_h^{j+1}) &:= \bar{\Sigma}'_{u^j,\lambda^{j+1}}(s_h^j,\lambda_h^{j+1})(\Delta \lambda_h^{j+1}),\\
\bar{\Sigma}'_{z^j,s^j}(\Delta s_h^j) &:= \bar{\Sigma}'_{z^j,s^j}(s_h^j,\lambda_h^{j+1})(\Delta s_h^j),\\
\bar{\Sigma}'_{z^j,\lambda^{j+1}}(\Delta \lambda_h^{j+1}) &:= \bar{\Sigma}'_{z^j,\lambda^{j+1}}(s_h^j,\lambda_h^{j+1})(\Delta \lambda_h^{j+1}).
\end{aligned}$$

The directional derivatives of the solution operators can be interpreted in terms of Lemma 6.1, and their discretizations are obtained accordingly for the discretized equations.

Lemma 6.1. *On I_j, $j = 0,\dots,m-1$ we consider the two point boundary value problems:*

$$\begin{aligned}
((\partial_t w_s,\varphi))_j + a'_u(u^j)(w_s,\varphi) + b'_q(q^j)(x_s,\varphi) + (w_s(\tau_j)-\rho,\varphi(\tau_j)) &= 0 \quad \forall \varphi \in X^j,\\
-((\partial_t y_s,\psi))_j + a''_{uu}(u^j)(w_s,\psi,z^j) + a'_u(u^j)(\psi,y_s) + (y_s(\tau_{j+1}),\psi(\tau_{j+1}))&\\
-\alpha_1 J_1^{j\prime\prime}(u^j)(w_s,\psi) &= 0 \quad \forall \psi \in X^j,\\
b''_{qq}(q^j)(x_s,\chi,z^j) + b'_q(q^j)(\chi,y_s) - \alpha_3(x_s,\chi)_{Q^j} &= 0 \quad \forall \chi \in Q^j.
\end{aligned}$$

and

$$\begin{aligned}
((\partial_t w_\lambda,\varphi))_j + a'_u(u^j)(w_\lambda,\varphi) + b_q(q^j)(x_\lambda,\varphi) + (w_\lambda(\tau_j),\varphi(\tau_j)) &= 0 \quad \forall \varphi \in X^j,\\
-((\partial_t y_\lambda,\psi))_j + a''_{uu}(u^j)(w_\lambda,\psi,z^j) + a'_u(u^j)(\psi,y_\lambda) + (y_\lambda(\tau_{j+1})-\xi,\psi(\tau_{j+1}))&\\
-\alpha_1 J_1^{j\prime\prime}(u^j)(w_\lambda,\psi) &= 0 \quad \forall \psi \in X^j,\\
b''_{qq}(q^j)(x_\lambda,\chi,z^j) + b'_q(q^j)(\chi,y_\lambda) - \alpha_3(x_\lambda,\chi)_{Q^j} &= 0 \quad \forall \chi \in Q^j.
\end{aligned}$$

Then the following identity holds for the solution operators:

$$\bar{\Sigma}'_{u^j,s^j} : \begin{cases} H \to H,\\ \rho \mapsto w_s(\tau_{j+1}) \end{cases} \quad \textit{and} \quad \bar{\Sigma}'_{z^j,s^j} : \begin{cases} H \to H,\\ \rho \mapsto y_s(\tau_j) \end{cases},$$

$$\bar{\Sigma}'_{u^j,\lambda^{j+1}} : \begin{cases} H \to H,\\ \xi \mapsto w_\lambda(\tau_{j+1}) \end{cases} \quad \textit{and} \quad \bar{\Sigma}'_{z^j,\lambda^{j+1}} : \begin{cases} H \to H,\\ \xi \mapsto y_\lambda(\tau_j) \end{cases}.$$

Proof. The statements of the lemma follow directly by differentiation of the boundary value problem (4.8) with respect to the boundary values and from the definition of the solution operators $\bar{\Sigma}_{u^j}$ and $\bar{\Sigma}_{z^j}$ for $j = 0,\dots,m-1$. □

Remark 6.2. Lemma 6.1 implies that in the context of Newton's method *additional intervalwise linear boundary value problems* have to be solved for the calculation of the directional derivatives of the state variables.

Let us assume that according to Remark 2.4 the functional J_2 has the special structure $J_2(s^m) = \frac{1}{2}\|s^m - \bar{u}(T)\|^2$. The general case for arbitrary J_2 follows analogously but is more

complicated with respect to the notation. For reasons of brevity, we introduce the following notation:

$$\begin{aligned}
A_j &:= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}, \quad j = 0, \dots m-1, \\
B_j &:= \begin{pmatrix} B_j^{11} & B_j^{12} \\ B_j^{21} & B_j^{22} \end{pmatrix} := \begin{pmatrix} B_j^{:1} & B_j^{:2} \end{pmatrix} := \begin{pmatrix} 0 & -\bar{\Sigma}'_{z^j,s^j} \\ 0 & -\bar{\Sigma}'_{u^j,s^j} \end{pmatrix}, \quad j = 0, \dots m-1, \\
C_j &:= \begin{pmatrix} C_j^{11} & C_j^{12} \\ C_j^{21} & C_j^{22} \end{pmatrix} := \begin{pmatrix} C_j^{:1} & C_j^{:2} \end{pmatrix} := \begin{pmatrix} -\bar{\Sigma}'_{z^j,\lambda^{j+1}} & 0 \\ -\bar{\Sigma}'_{u^j,\lambda^{j+1}} & 0 \end{pmatrix}, \quad j = 0, \dots m-1, \\
F &:= \begin{pmatrix} 0 & -I \end{pmatrix}, \\
r_I &:= s_h^0 - u_0, \\
r_j &:= \begin{pmatrix} s_h^{j+1} - \bar{\Sigma}_{u^j}(s_h^j, \lambda_h^{j+1}) \\ \lambda_h^j - \bar{\Sigma}_{z^j}(s_h^j, \lambda_h^{j+1}) \end{pmatrix}, \quad j = 0, \dots m-1, \\
r_F &:= \lambda_h^m - (s_h^m - \bar{u}).
\end{aligned} \tag{6.2}$$

$$\tag{6.3}$$

Here, $B_j^{:1}$, $B_j^{:2}$, $C_j^{:1}$, and $C_j^{:2}$ denote the columns of the pseudo matrices. The linearized system of matching conditions (6.1) can now be written in short pseudo matrix notation:

$$\underbrace{\begin{pmatrix} I & & & & & & & \\ B_0^{:2} & A_0 & C_0 & & & & & \\ & B_1 & A_1 & C_1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & B_{m-2} & A_{m-2} & C_{m-2} & & \\ & & & & B_{m-1} & A_{m-1} & C_{m-1}^{:1} & \\ & & & & & F & I \end{pmatrix}}_{=:K} \underbrace{\begin{pmatrix} \Delta s_h^0 \\ (\Delta\lambda_h^0, \Delta s_h^1)^T \\ (\Delta\lambda_h^1, \Delta s_h^2)^T \\ \vdots \\ (\Delta\lambda_h^{m-2}, \Delta s_h^{m-1})^T \\ (\Delta\lambda_h^{m-1}, \Delta s_h^m)^T \\ \Delta\lambda_h^m \end{pmatrix}}_{=:\Delta x} = - \underbrace{\begin{pmatrix} r_I \\ r_0 \\ r_1 \\ \vdots \\ r_{m-2} \\ r_{m-1} \\ r_F \end{pmatrix}}_{=:b}. \tag{6.4}$$

The solution of the multiple shooting problem by application of Newton's method to the system of matching condition reads consequently in algorithmic formulation as presented in Algorithm 6.1.

Algorithm 6.1 Newton's method for the indirect approach

1: Set $i = 0$.
2: Start with initial guess for multiple shooting variables $x_i = (s_{h,i}^0, \dots, s_{h,i}^m, \lambda_{h,i}^0, \dots, \lambda_{h,i}^m)^T$.
3: **repeat**
4: Calculate the residuals $b_i = (r_{I,i}, r_{0,i}, \dots, r_{m-1,i}, r_{F,i})^T$ according to (6.3).
5: Solve the linear system (6.4) for the Newton update,

$$K_i \Delta x_i = -b_i, \quad \text{with} \quad \Delta x_i = (\Delta s_{h,i}^0, \dots, \Delta s_{h,i}^m, \Delta\lambda_{h,i}^0, \dots \lambda_{h,i}^m)^T.$$

6: Calculate next iterate

$$x_{i+1} = x_i + \Delta x_i.$$

7: Set $i = i + 1$.
8: **until** $\|b_i\|_{\tilde{H}\times\tilde{H}} := \sqrt{\|r_{I,i}\|^2 + \sum_{j=0}^{m-1} \|r_{j,i}\|^2 + \|r_{F,i}\|^2} < \text{tol}$.

Remark 6.3. (Matrix-vector based formulation of Newton's method) The consideration of Newton's method in the discrete case needs the determination of the coefficient vectors of the solutions from a linear system of equations. Therefore, we need the concretization of the linear system (6.4) for given bases of the discretized spaces on the multiple shooting nodes. We do this exemplarily for only one block row $(B_j \quad A_j \quad C_j)$ of the pseudo block matrix K. We memorize the corresponding equations of the linearized system (6.1):

$$\begin{aligned}(\Delta\lambda_h^j - \bar{\Sigma}'_{z^j,s^j}(\Delta s_h^j) - \bar{\Sigma}'_{z^j,\lambda^{j+1}}(\Delta\lambda_h^{j+1}), v) &= -(\lambda_h^j - \bar{\Sigma}_{z^j}(s_h^j, \lambda_h^{j+1}), v) \qquad \forall v \in H_h^j,\\ (\Delta s_h^{j+1} - \bar{\Sigma}'_{u^j,s^j}(\Delta s_h^j) - \bar{\Sigma}'_{u^j,\lambda^{j+1}}(\Delta\lambda_h^{j+1}), v) &= -(s_h^{j+1} - \bar{\Sigma}_{u^j}(s_h^j, \lambda_h^{j+1}), v) \qquad \forall v \in H_h^{j+1}.\end{aligned}$$

Now, we assume that $\varphi_0^j, \ldots, \varphi_{n_j}^j$ is a basis of H_h^j, and the coefficient vectors of the Newton increments Δs_h^j and $\Delta\lambda_h^j$ with respect to this basis are given by $\Delta\xi^{j,s} \in \mathbb{R}^{n_j}$ and $\Delta\xi^{j,\lambda} \in \mathbb{R}^{n_j}$. With this notation at hand, we are able to rewrite the linear equations as

$$\begin{aligned}G_0\Delta\xi^{j,\lambda} - G_1^s\Delta\xi^{j,s} - G_1^\lambda\Delta\xi^{j+1,\lambda} &= -d_0,\\ G_2\Delta\xi^{j+1,s} - G_3^s\Delta\xi^{j,s} - G_3^\lambda\Delta\xi^{j+1,\lambda} &= -d_1,\end{aligned}$$

where the matrices $G_0, G_1^s \in \mathbb{R}^{n_j\times n_j}$, $G_1^\lambda \in \mathbb{R}^{n_j\times n_{j+1}}$, $G_2, G_3^\lambda \in \mathbb{R}^{n_{j+1}\times n_{j+1}}$, $G_3^s \in \mathbb{R}^{n_{j+1}\times n_j}$ have to be understood according to the following identities:

$$\begin{aligned}(G_0)_{li} &= (\varphi_i^j, \varphi_l^j), & (G_1^s)_{li} &= (\bar{\Sigma}'_{z^j,s^j}(\varphi_i^j), \varphi_l^j), & (G_1^\lambda)_{li} &= (\bar{\Sigma}'_{z^j,\lambda^{j+1}}(\varphi_i^{j+1}), \varphi_l^j),\\ (G_2)_{li} &= (\varphi_i^{j+1}, \varphi_l^{j+1}), & (G_3^s)_{li} &= (\bar{\Sigma}'_{u^j,s^j}(\varphi_i^j), \varphi_l^{j+1}), & (G_3^\lambda)_{li} &= (\bar{\Sigma}'_{u^j,\lambda^{j+1}}(\varphi_i^{j+1}), \varphi_l^{j+1})\end{aligned}$$

and the right-hand side vectors $d_0 \in \mathbb{R}^{n_j}$ and $d_1 \in \mathbb{R}^{n_{j+1}}$ according to

$$\begin{aligned}(d_0)_l &= (\lambda_k^j - \bar{\Sigma}_{z^j}(s_k^j, \lambda_k^{j+1}), \varphi_l^j),\\ (d_1)_l &= (s_k^{j+1} - \bar{\Sigma}_{u^j}(s_k^j, \lambda_k^{j+1}), \varphi_l^{j+1}).\end{aligned}$$

We remark, that d_0 and d_1 are usually different from the coefficient vectors for the right-hand sides. With this interpretation at hand, the transfer of the function space method to a matrix-vector based one is straightforward by setting up the discrete linear system for the determination of the coefficient vectors of the Newton updates. In addition, it is often convenient and suitable to replace the Hilbert space norm on H by the l^2-norm on the finite dimensional vector space $\mathbb{R}^{n_j}$.

For reasons of vividness, let us consider Newton's method for Example 2.3:

Example 6.1. (Newton's method for Example 2.3) Reconsidering the multiple shooting approach for Example 2.3 which was introduced in Example 4.1, we retrieve the following linearized system:

$$\begin{aligned}(\Delta s_h^0, \varphi) &= -(s_h^0 - u_0, \varphi) && \forall\varphi \in H_h^0,\\ (\Delta\lambda_h^0 - \bar{\Sigma}'_{z^0,s^0}(\Delta s_h^0), \varphi) &= -(\lambda_h^0 - \bar{\Sigma}_{z^0}(s_h^0), \varphi) && \forall\varphi \in H_h^0,\\ (\Delta s_h^1 - \bar{\Sigma}'_{u^0,\lambda^1}(\Delta\lambda_h^1), \varphi) &= -(s_h^1 - \bar{\Sigma}_{u^0}(\lambda_h^1), \varphi) && \forall\varphi \in H_h^1,\\ (\Delta\lambda_h^1 - \bar{\Sigma}'_{z^1,s^1}(\Delta s^1), \varphi) &= -(\lambda_h^1 - \bar{\Sigma}_{z^1}(s_h^1), \varphi) && \forall\varphi \in H_h^1,\\ (\Delta s_h^2 - \bar{\Sigma}'_{u^1,\lambda^2}(\Delta\lambda_h^2), \varphi) &= -(s^2 - \bar{\Sigma}_{u^1}(\lambda_h^2), \varphi) && \forall\varphi \in H_h^2,\\ (\Delta\lambda_h^2, \varphi) &= -(\lambda_h^2, \varphi) && \forall\varphi \in H_h^2.\end{aligned}$$

Differentiation of the (undiscretized) boundary value problems (4.15) yields for the derivatives of the solution operators the following identities:

On I_j let the boundary value problem be given as follows: for all $\varphi, \psi, \chi \in X$:

$$\begin{aligned}((\partial_t w, \varphi))_j + ((\nabla w, \nabla \varphi))_j + ((3(u^j)^2 w, \varphi))_j + (w(\tau_j), \varphi(0)) &= ((x, \varphi))_j, \\ -((\partial_t y, \psi))_j + ((\nabla y, \nabla \psi))_j + ((3(u^j)^2 y, \psi))_j + (y(\tau_{j+1}) - \xi, \psi(T)) &= ((w, \psi))_j - ((6u^j w z^j, \psi))_j, \\ ((x, \chi))_j &= -((y, \chi))_j.\end{aligned}$$

Then

$$\bar{\Sigma}'_{u^j, \lambda^{j+1}}(\xi) = w(\tau_{j+1}).$$

For all $\varphi, \psi, \chi \in X$:

$$\begin{aligned}((\partial_t w, \varphi))_j + ((\nabla w, \nabla \varphi))_j + ((3(u^j)^2 w, \varphi))_j + (w(\tau_j) - \rho, \varphi(0)) &= ((x, \varphi))_j, \\ -((-\partial_t y, \psi))_j + ((\nabla y, \nabla \psi))_j + ((3(u^j)^2 y, \psi))_j + (y(\tau_{j+1}), \psi(T)) &= ((w, \psi))_j - ((6u^j w z^j, \psi))_j, \\ ((x, \chi))_j &= -((y, \chi))_j.\end{aligned}$$

Then

$$\bar{\Sigma}'_{z^j, s^j}(\rho) = y(\tau_j).$$

For the discrete derivatives of the solution operators this holds analogously for the discretized equations.

Throughout this chapter, we consider this example for pointing out the setup of the different methods. In the following, we discuss the solution of the linearized system (6.4) by application of an iterative solver from the class of Krylov subspace methods.

6.1.2 The GMRES Method for the Solution of the Linearized System

The pseudo matrix notation of problem (6.4) reveals a *sparse tridiagonal block structure* of the problem with identity operators on the diagonal. This property proves useful when considering the solution of the linearized system by application of a *preconditioned Krylov subspace method*. The solution of a linearized system with a similar structure as in (6.4) is extensively studied in [20] and [14], where the application of a preconditioned Krylov subspace method is suggested. Our choice is a *preconditioned generalized minimum residual method* (GMRES), and as a reasonable preconditioner the *forward backward block Gauss Seidel* preconditioner is chosen. To underline on the one hand the necessity of a preconditioner and on the other hand the advantage of forward backward block Gauss Seidel preconditioning, different preconditioners have been tested at a linear example. This example presents the simplest case of an optimal control problem constrained by a parabolic PDE and serves quite well for the outline of the basic properties of the GMRES method.

Example 6.2. Let $\Omega = (-1, 1) \times (-1, 1)$ be a square shaped domain, $I = (0, 5)$, and let the optimal control problem be given by

$$\min_{(q,u) \in Q \times X} J(q, u) := \frac{1}{2} \|u(T) - 0.5\|^2_{L^2(\Omega)} + \frac{0.001}{2} \int_I \|q(t)\|^2_{L^2(\Omega)} dt$$

subject to the constraining heat equation

$$\begin{aligned} \partial_t u - \Delta u &= q && \text{in} && \Omega \times I, \\ u &= 0 && \text{on} && \partial\Omega \times I, \\ u &= \cos\left(\frac{\pi}{2}x\right)\cos\left(\frac{\pi}{2}y\right) && \text{in} && \Omega \times \{0\}\,. \end{aligned}$$

We consider the problem on a five times globally refined mesh, that is a mesh of 1024 cells. Due to the linearity of the problem, the outer Newton method converges in one step which results in only one application of the GMRES method for the solution of the linearized system of matching conditions. In Figure 6.2, we compare the number of iterative steps until convergence for different preconditioners and various numbers of intervals. The results of the calculations illustrate that preconditioning is essential. The forward backward block Gauss Seidel preconditioner yields by far the best results of the considered preconditioners, and additionally the number of GMRES iterations until convergence is approximately constant 77 steps for the different numbers of intervals. The assumption, that the forward backward Gauss Seidel preconditioner leaves the number of GMRES iterations constant for different time domain decompositions was substantiated by calculation of other examples not presented in this thesis.

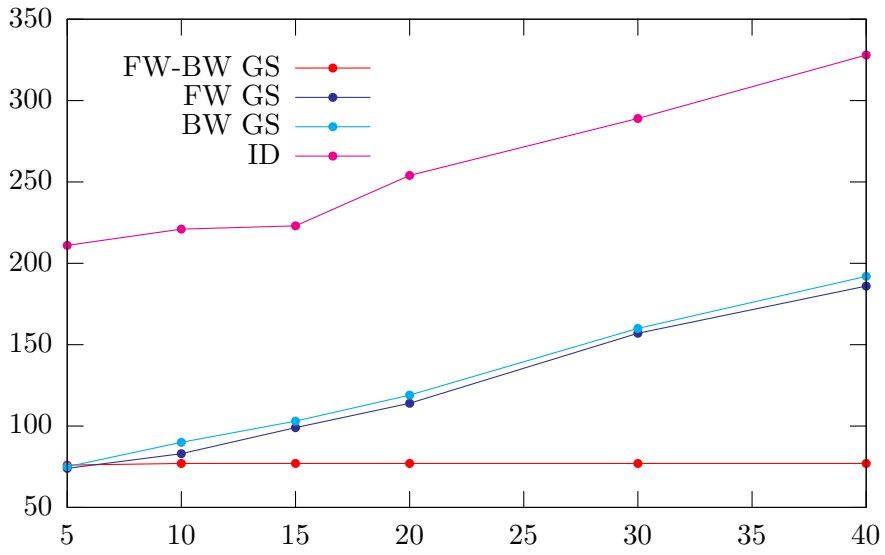


Figure 6.2: Results for the forward backward Gauss Seidel preconditioner (red), the forward Gauss Seidel preconditioner (dark blue), the backward Gauss Seidel preconditioner (blue) and without preconditioner (magenta) calculated for Example 6.2. (x-axis: number of intervals, y-axis: number of iterations)

Remark 6.4. (Parallelizability) As a crucial drawback of this approach we should mention that the preconditioner presented above is not parallelizable. As a matter of fact, several preconditioners have been studied in [14]. Numerical experiments show that so far considered

good preconditioners for problem (6.4) are not parallelizable while parallelizable preconditioners do not yield the desired improvement with respect to the convergence of the Krylov subspace method.

For the purpose of presenting the block Gauss Seidel preconditioner, the following pseudo block matrix decomposition into a lower triangular pseudo block matrix L, an upper triangular pseudo block matrix U and a diagonal pseudo block matrix D is introduced:

$$\begin{pmatrix} \ddots & & -U \\ & D & \\ -L & & \ddots \end{pmatrix} := \begin{pmatrix} I & & & & \\ B_0^{:2} & A & C_0 & & \\ & \ddots & \ddots & \ddots & \\ & & B_{m-1} & A & C_{m-1}^{:1} \\ & & & F & I \end{pmatrix}.$$

The preconditioning pseudo matrices for the forward block Gauss Seidel and the backward block Gauss Seidel preconditioner read

$$P_{\text{fwd}} := D - L \quad \text{and} \quad P_{\text{bwd}} := D - U,$$

and for the forward backward block Gauss Seidel preconditioner the preconditioning pseudo matrix is defined as

$$P := (D - L)D^{-1}(D - U). \tag{6.5}$$

Thus, the application of the forward backward block Gauss Seidel preconditioner in the GMRES method for the solution of (6.4) is equivalent to the solution of the system

$$P^{-1}Kx = P^{-1}b \tag{6.6}$$

by the unpreconditioned GMRES method. Due to the special structure of the matrices, and due to $D = D^{-1}$, the calculation of $P^{-1}v$, $v \in \tilde{H} \times \tilde{H}$, can be performed by forward and backward substitution as outlined in the following:

$$\underbrace{\begin{pmatrix} I & & & & & & \\ B_0^{:2} & A & & & & & \\ & B_1 & A & & & & \\ & & \ddots & \ddots & & & \\ & & & B_{m-2} & A & & \\ & & & & B_{m-1} & A & \\ & & & & & F & I \end{pmatrix}}_{D-L} \underbrace{\begin{pmatrix} y_0 \\ (x_0, y_1)^T \\ (x_1, y_2)^T \\ \vdots \\ (x_{m-2}, y_{m-1})^T \\ (x_{m-1}, y_m)^T \\ x_m \end{pmatrix}}_{z} = \underbrace{\begin{pmatrix} v_I \\ v_0 \\ v_1 \\ \vdots \\ v_{m-2} \\ v_{m-1} \\ v_F \end{pmatrix}}_{v}$$

$$\Leftrightarrow \begin{Bmatrix} y_0 = v_I \\ (x_0, y_1)^T = v_0 - B_0^{:2} y_0 \\ (x_1, y_2)^T = v_1 - B_2(x_0, y_1)^T \\ \vdots \\ (x_{m-2}, y_{m-1})^T = v_{m-2} - B_{m-2}(x_{m-3}, y_{m-2}^T) \\ (x_{m-1}, y_m)^T = v_{m-1} - B_{m-1}(x_{m-2}, y_{m-1})^T \\ x_m = v_F - F(x_{m-1}, y_m)^T \end{Bmatrix}.$$

The inversion of $D - U$ is treated analogously:

$$(D-U)z = v \Leftrightarrow \left\{ \begin{array}{c} x_m = v_F \\ (x_{m-1}, y_m)^T = v_{m-1} - C_{m-1}^{:1} x_m \\ (x_{m-2}, y_{m-1})^T = v_{m-2} - C_{m-2}(x_{m-1}, y_m)^T \\ \vdots \\ (x_1, y_2)^T = v_1 - C_1 (x_2, y_3)^T \\ (x_0, y_1)^T = v_0 - C_0^{2:} (x_1, y_2)^T \\ y_0 = v_I \end{array} \right\}.$$

Here, the pseudo matrix vector products, that is the evaluation of the directional derivatives, and the residuals are calculated according to Lemma 6.1 from intervalwise boundary value problems. The (unpreconditioned) GMRES method (see for instance the textbook of Saad [34]) is proposed in Algorithm 6.2. Therefore, in addition to the already introduced pseudo vectors $\Delta x, b \in \tilde{H}_h \times \tilde{H}_h$, we redefine $r \in \tilde{H}_h \times \tilde{H}_h$ as the residual of the linearized system, and introduce additional auxiliary pseudo vectors $v, w \in \tilde{H}_h \times \tilde{H}_h$ and real numbers $\beta, y_i, z_i, h_{il}, \tau, \nu, c_i, c_s \in \mathbb{R}$ for the formulation of the algorithm.

Algorithm 6.2 GMRES for the solution of (6.4)

Chose tol.
Set $i = -1$.
Start with initial guess for the Newton update Δx_0.
Calculate $r_0 := b_0 - K\Delta x_0$, $\beta := \|r_0\|$, $v_1 := {}^{r_0}/_{\beta}$, $z_0 := \beta$.
repeat
    $i = i + 1$.
    $w_i = K v_i$.
    for $l = 0$ **to** i **do**
        Set $h_{li} := (v_l, w_i)_{\tilde{H} \times \tilde{H}}$.
        Set $w_i := w_i - h_{li} v_l$.
    Set $h_{i+1,i} := \|w_i\|_{\tilde{H} \times \tilde{H}}$.
    Set $v_{i+1} := {}^{w_i}/_{h_{i+1,i}}$.
    for $l = 0$ **to** $i - 1$ **do**
        Set $\begin{pmatrix} h_{li} \\ h_{l+1,i} \end{pmatrix} := \begin{pmatrix} c_l & s_l \\ -s_l & s_l \end{pmatrix} \begin{pmatrix} h_{li} \\ h_{l+1,i} \end{pmatrix}$.
    Set $\tau := |h_{ii}| + |h_{i,i+1}|$.
    Set $\nu := \tau \cdot \sqrt{(h_{ii}/\tau)^2 + (h_{i,i+1}/\tau)^2}$.
    Set $c_i := h_{ii}/\nu$, and $s_i := h_{i,i+1}/\nu$.
    Set $z_{i+1} := -s_i z_i$. Set $z_i := -c_i z_i$.
until ${}^{|z_i|}/_{\beta} <$ tol.
$y_i := {}^{z_i}/_{h_{ii}}$.
for $l = i - 1$ **down to** 0 **do**
    Set $y_l = {}^{(z_l - \sum_{j=l+1}^{i} h_{lj} y_j)}/_{h_{ll}}$.
$\Delta x_i = \Delta x_0 + \sum_{l=0}^{i} y_l v_l$.

Remark 6.5. (Matrix-vector based formulation of the GMRES method) As before, the practical implementation of the method is based on the determination of the coefficient vectors of the solutions. The interpretation of those is due to Remark (6.3) and thus not repeated particularly for this method.

From the discussion above, it has become clear that all preconditioners can only be applied at additional costs. In detail, for the forward backward block Gauss Seidel preconditioner we have to solve 2 additional linear boundary value problems on each interval, which makes a total amount of 4 linear boundary value problems per interval for every iteration of the GMRES method. Over all, most of the effort is spend for the preconditioned iterative solver, outlined by a simple analysis of Example 6.1. In Figure 6.3 the total number of solved linear and nonlinear problems is presented for the different number of intervals. We have previously seen, that for this example the preconditioned iteration needs 77 iterative steps, which makes a total number of $4 \cdot 77$ linear boundary value problems per interval, or over all $m \cdot 308$ problems until convergence. On the other hand, we have to evaluate the matching conditions twice, such that $2 \cdot m$ nonlinear boundary value problems have to be solved. Half of the solved linear boundary value problems are due to the preconditioning, the other half are solved during the calculation of the directional derivatives themselves. Nevertheless, we see that despite the additional effort, the preconditioned iteration is by far better than the unpreconditioned GMRES method.

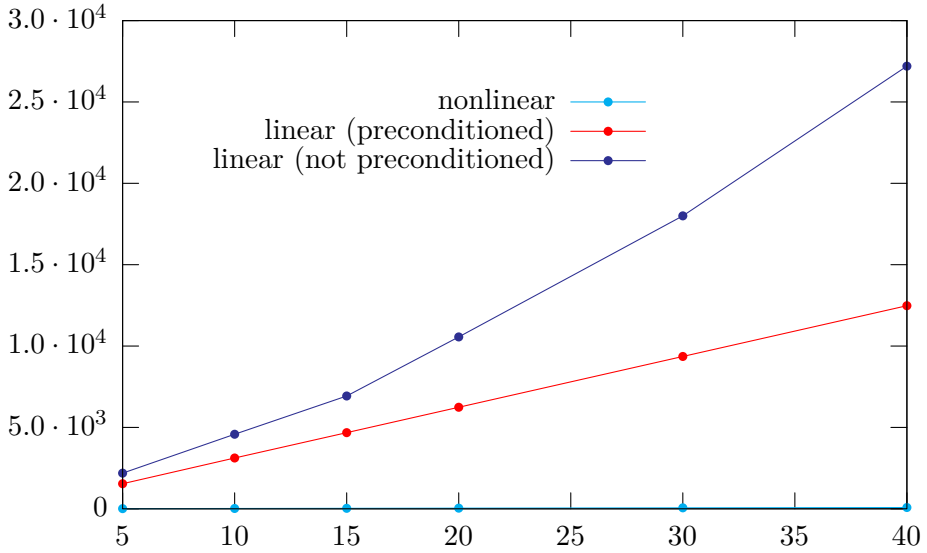


Figure 6.3: Number of solved linear and nonlinear boundary value problems for different numbers of shooting intervals, calculated with and without preconditioning. (x-axis: number of intervals, y-axis: number of solved problems)

Remark 6.6. We should keep in mind that for different number of intervals, the interval length, and thus the solution time for the intervalwise boundary value problems, is different, too. Therefore, Figure 6.3 contains only information on the relative efficiency of the preconditioners, but not on the absolute effort and time needed for the performance of the methods.

Next, we investigate efficient solution techniques for the intervalwise boundary value problems, linear and nonlinear ones. This effort has resulted in two different approaches which we explain in the following paragraphs.

6.1.3 Solution of the Interval Problems – Newton's method

The calculation of the Newton residuals in (6.1) for the outer Newton method requires the solution of the intervalwise nonlinear boundary value problems (5.6). In this subsection we present a solution strategy which is motivated by the idea of *linearizing the nonlinear boundary value problems* within an *inner Newton method*. It has the great advantage of solving both linear and nonlinear boundary value problems by only one solver. Therefore, the whole problem is brought down to the solution of linear boundary value problems by standard techniques. We sketch this idea in the flow diagram in Figure 6.4.

By application of Newton's method on the nonlinear boundary value problems we obtain another set of linear boundary value problems of the same structure as those presented in Lemma 6.1.

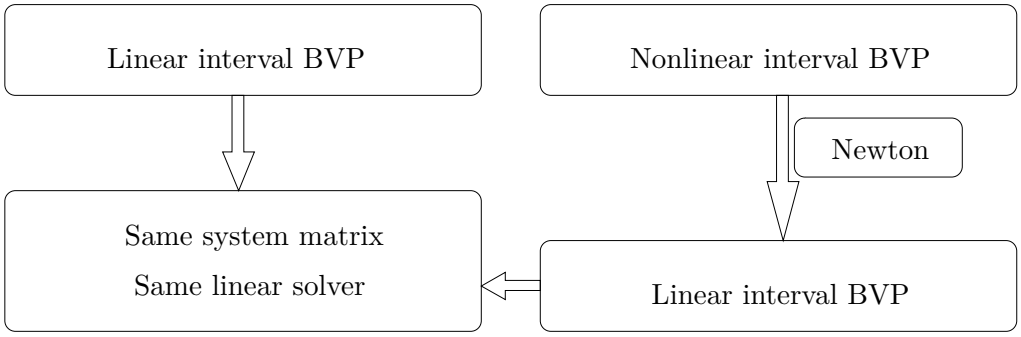


Figure 6.4: Transformation of the nonlinear boundary value problems.

Remark 6.7. We set up this approach for the undiscretized problem only and remark, that in the discretized case the corresponding algorithm is obtained by replacing the undiscretized equations, solutions, and spaces by their discrete equivalents.

We recall the formulation of the intervalwise nonlinear boundary value problems in the following, before we proceed with the description of Newton's method for their solution:

Primal equation:

$$((\partial_t u^j, \varphi))_j + a(u^j)(\varphi) + b(q^j)(\varphi) + (u^j(\tau_j) - s^j, \varphi(\tau_j)) - ((f, \varphi))_j = 0 \quad \forall \varphi \in X^j. \tag{4.8a}$$

Dual equation:

$$-((\partial_t z^j, \psi))_j + a'_u(u^j)(\psi, z^j) + (z^j(\tau_{j+1}) - \lambda^{j+1}, \psi(\tau_{j+1})) - \alpha_1 J_1^{j\prime}(u^j)(\psi) = 0 \quad \forall \psi \in X^j. \tag{4.8b}$$

Control equation:

$$b'_q(q^j)(\chi, z^j) - \alpha_3 (q^j, \chi)_{Q^j} = 0 \quad \forall \chi \in Q^j. \tag{4.8c}$$

The application of Newton's method to the whole nonlinear system results in the linearized system (6.8) from which the Newton update $(\delta q_l^j, \delta u_l^j, \delta z_l^j) \in Q^j \times X^j \times X^j$ in step l is calculated.

Primal equation:

$$(\!(\partial_t \delta u_l^j, \varphi)\!)_j + a_u'(u_l^j)(\delta u_l^j, \varphi) + b_q'(q_l^j)(\delta q_l^j, \varphi) + (\delta u_l^j(\tau_j), \varphi(\tau_j)) = -(\!(r_u, \varphi)\!)_j \quad \forall \varphi \in X^j. \tag{6.8a}$$

Dual equation:

$$\begin{aligned} -(\!(\partial_t \delta z_l^j, \psi)\!)_j + a_{uu}''(u^j)(\delta u_l^j, \psi, z^j) + a_u'(u^j)(\psi, \delta z_l^j) + (\delta z_l^j(\tau_{j+1}), \psi(\tau_{j+1})) \\ - \alpha_1 J_1^{j\prime\prime}(u_l^j)(\delta u_l^j, \psi) = -(\!(r_z, \psi)\!)_j \quad \forall \psi \in X^j. \end{aligned} \tag{6.8b}$$

Control equation:

$$b_{qq}''(q_l^j)(\delta q_l^j, \chi, z_l^j) + b_q'(q_l^j)(\chi, \delta z_l^j) - \alpha_3(\delta q_l^j, \chi)_{Q^j} = -(r_q, \chi)_{Q^j} \quad \forall \chi \in Q^j. \tag{6.8c}$$

Here, the right-hand side is given by the residual of the nonlinear boundary value problem

$$\begin{aligned} (\!(r_u^j, \varphi)\!)_j &:= (\!(\partial_t u_l^j, \varphi)\!)_j + a(u_l^j)(\varphi) + b(q_l^j)(\varphi) + (u_l^j(\tau_j) - s^j, \varphi(\tau_j)) - (\!(f, \varphi)\!)_j, \\ (\!(r_z^j, \psi)\!)_j &:= -(\!(\partial_t z_l^j, \psi)\!)_j + a_u'(u_l^j)(\psi, z_l^j) + (z_l^j(\tau_{j+1}) - \lambda^{j+1}, \psi(\tau_{j+1})) - \alpha_1 J_1^{j\prime}(u^j)(\psi), \\ (r_q^j, \chi)_{Q^j} &:= b_q'(q_l^j)(\chi, z_l^j) - \alpha_3(q_l^j, \chi)_{Q^j}. \end{aligned} \tag{6.9}$$

The subsequent Newton iterate is then calculated as usual by updating

$$q_{l+1}^j = q_l^j + \nu_l \delta q_l^j, \quad u_{l+1}^j = u_l^j + \nu_l \delta u_l^j, \quad z_{l+1}^j = z_l^j + \nu_l \delta z_l^j,$$

where the damping parameter $\nu_l \in (0, 1]$ is determined by standard backtracking techniques such that a descent in the Newton direction is guaranteed. The pseudo code notation of the algorithm is given in Algorithm 6.3.

Algorithm 6.3 Newton's method for the solution of (4.8)

1: Set $l = 0$.
2: Start with an initial guess $q_0^j \in Q^j$, $u_0^j \in X^j$, $z_0^j \in X^j$.
3: **repeat**
4: Calculate the residuals $r_{u,l}^j$, $r_{z,l}^j$ and $r_{q,l}^j$ according to (6.9).
5: Solve the linear boundary value problem (6.8) for δq_l^j, δu_l^j, δz_l^j.
6: Choose ν_l such that for the next iterate a descent in the residual is obtained.
7: Calculate next iterate

$$(q_{l+1}^j, u_{l+1}^j, z_{l+1}^j) = (q_l^j, u_l^j, z_l^j) + \nu_l(\delta q_l^j, \delta u_l^j, \delta z_l^j).$$

8: Set $l = l + 1$.
9: **until** $\sqrt{\|r_{u,l}^j\|^2 + \|r_{z,l}^j\|^2 + \|r_{q,l}^j\|_Q^2} < \text{tol}$

A comparison shows that the linear boundary value problems developed above coincide with those for the directional derivatives of the states in Lemma 6.1 except for the right-hand

sides and the initial values. As a convenient consequence, we are able to apply the same linear solver for both linear and nonlinear boundary value problems.

Summarizing, this approach requests only the solution of interval wise linear boundary value problems instead of nonlinear ones, and is quite simple with respect to the implementation. In what follows, we discuss the solution of the linear interval wise boundary value problems. Further on, we give some results on the performance of the approach and point out the limitations and difficulties that occur.

6.1.4 Solution of the Linear Problems – Fixed Point Iteration and Gradient Method

Investigating the structure of the linear boundary value problems, the obvious and easiest way of solving would be the application of a *modified fixed point iteration*. The basic idea of a fixed point iteration on I_j is to evaluate u^j, z^j and q^j successively in a loop: As an example, in step l of the fixed point iteration, given a control q_l^j, we calculate the state u_l^j from the primal equation. The dual state z_l^j can then be calculated from the dual equation and an update for the control, q_{l+1}^j is obtained from the control equation. Unfortunately, numerical experiments show that the convergence behavior of a fixed point iteration is extremely bad for fully coupled boundary value problems. A closer investigation reveals that in this context, it is *equivalent* to a *gradient method with step size* $1/\alpha_3$ as we prove in the following.

Remark 6.8. Again, the methods are derived for the undiscretized problem only and the extension to the discrete case is obtained straightforward by replacing the undiscretized equations, solutions, solution operators, functionals and spaces through their discrete equivalents.

For the purpose of showing the equivalence of a fixed point iteration and the gradient method, it is sufficient to consider a simplified optimal control problem. The proof for the general case of an arbitrary optimal control problem follows analogously with the same basic ideas but needs a more complicated notational framework and is quite hard to follow up.

For the ease of presentation we omit the interval index j. We consider on $I = (0, T)$ the optimization problem

$$\min_{q,u} J(q,u) := \frac{\alpha_1}{2}\int_I \|u(t) - \bar{u}(t)\|^2 dt + \frac{\alpha_2}{2}\|u(T) - \bar{u}_T\|^2 + \frac{\alpha_3}{2}\|q(t)\|_Q^2 \tag{6.10a}$$

subject to

$$((\partial_t u, \varphi)) + a(u,\varphi) + b(q,\varphi) + (u(0), \varphi(0)) = 0 \quad \forall \varphi \in X. \tag{6.10b}$$

The choice of a homogenous initial value and right-hand side is suitable due to the fact, that for the linear constraining equation the general case follows immediately by an affine linear shift of the solution space. The Lagrangian is defined as before by

$$\mathcal{L}(q,u,z) = J(q,u) - \{((\partial_t u, z)) + a(u,z) + b(q,z) + (u(0), z(0))\}, \tag{6.11}$$

and the corresponding optimality system is given by the linear boundary value problem stated below.

Primal equation:

$$((\partial_t u, \varphi)) + a(u, \varphi) + b(q, \varphi) + (u(0), \varphi(0)) = 0 \quad \forall \varphi \in X.$$

Dual equation:

$$-((\partial_t z, \psi)) + a(\psi, z) + (z(T) - \alpha_2(u(T) - \bar{u}_T), \psi(T)) - \alpha_1((u - \bar{u}, \psi)) = 0 \quad \forall \psi \in X.$$

Control equation:

$$\alpha_3(q, \chi)_Q - b(\chi, z) = 0 \quad \forall \chi \in Q.$$

Again, we introduce the solution operators for the constraining equation (6.10b)

$$S : \begin{cases} Q \to X \\ q \mapsto u \end{cases} \quad \text{and} \quad \bar{S} : \begin{cases} Q \to H \\ q \mapsto u(T) \end{cases}.$$

We reformulate problem (6.10) in the so called reduced formulation by replacing u in (6.10a) by Sq and $u(T)$ by $\bar{S}q$:

$$\min_q j(q) := J(q, Sq) = \frac{\alpha_1}{2} \int_I \|Sq(t) - \bar{u}(t)\|^2 dt + \frac{\alpha_2}{2} \|\bar{S}q - \bar{u}_T\|^2 + \frac{\alpha_3}{2} \|q(t)\|_Q^2. \tag{6.13}$$

A more concrete description of the reduced approach is given later on in Section 6.1.6.

The gradient method in function space with step size $1/\alpha_3$ for this problem is described by the formula

$$(q_{\text{new}}, \delta q)_Q = (q_{\text{old}}, \delta q)_Q - \frac{1}{\alpha_3} j'(q_{\text{old}})(\delta q) \quad \forall \delta q \in Q.$$

We proceed with the development of a simplified representation of the gradient. From the definition of the reduced cost functional in (6.13) and the Lagrangian in (6.11) we obtain for given q and $u = Sq$ the identity

$$j(q) = \mathcal{L}(q, u, z).$$

Now, differentiation with respect to q into the direction $\delta q \in Q$ yields the following representation of the gradient $j'(q)(\delta q)$:

$$j'(q)(\delta q) = \mathcal{L}'_q(q, u, z)(\delta q) + \mathcal{L}'_u(q, u, z)(\delta u) + \mathcal{L}'_z(q, u, z)(\delta z)$$

where $\delta u = \frac{du}{dq}(\delta q)$ and $\delta z = \frac{dz}{dq}(\delta q)$.

Due to $u = Sq$, the primal equation is fulfilled, and thus $\mathcal{L}'_z(q, u, z)(\delta z) = 0$. Choosing z such that the dual equation is fulfilled, too, we obtain additionally $\mathcal{L}'_u(q, u, z)(\delta u) = 0$ and retrieve for the gradient

$$\begin{aligned} j'(q)(\delta q) &= \mathcal{L}'_q(q, u, z)(\delta q) \\ &= \alpha_3(q, \delta q)_Q - b(\delta q, z). \end{aligned} \tag{6.14}$$

Now, we have everything at hand to investigate the relation between a fixed point iteration and the gradient method. We give the pseudo code notation for one step of both iterations, and a step-by-step comparison immediately gives evidence of their equivalence.

Algorithm 6.4 One step of the gradient method for a linear BVP

Require: q_{old} given.

1: Calculate u_{old} as

$$u_{\text{old}} = Sq_{\text{old}}.$$

2: Solve dual equation for z_{old}.

3: Evaluate gradient according to

$$j'(q)(\delta q) = \alpha_3(q_{\text{old}}, \delta q)_Q - b(\delta q, z_{\text{old}}).$$

4: Obtain new control q_{new} as

$$\begin{aligned}(q_{\text{new}}, \delta q)_Q &= (q_{\text{old}}, \delta q) - \frac{1}{\alpha_3}(\alpha_3(q_{\text{old}} - b(\delta q, z_{\text{old}}), \delta q)_Q) \\ &= \frac{1}{\alpha_3} b(\delta q, z_{\text{old}}).\end{aligned}$$

Algorithm 6.5 One step of the fixed point iteration for a linear BVP

Require: q_{old} given.

1: Calculate u_{old} as

$$u_{\text{old}} = Sq_{\text{old}}.$$

2: Solve dual equation for z_{old}.

3: Obtain new control q_{new} from the control equation

$$(q_{\text{new}}, \delta q)_Q = \frac{1}{\alpha_3} b(\delta q, z_{\text{old}}) = 0 \quad \forall \delta q \in Q.$$

Comparing the steps of both methods, the equivalence is quite obvious. The first three steps are the same for both algorithms, and step four and five of the first algorithm correspond to the fourth step of the second one. The bad convergence behavior of the fixed point iteration can thus be explained by the corresponding step size of the gradient method. For a small regularization parameter $\alpha_3 > 0$, the step size $1/\alpha_3$ becomes extremely large and convergence is no longer guaranteed.

A promising alternative for the solution of the boundary value problems is the application of a conjugate gradient method (CG method). The description of this method for the function space problem differs from the formulation for finite dimensional matrix vector systems as the equivalent to the usual matrix vector products is given by the application of operators and their duals on functions in Q. The CG method for the solution of the linear boundary value problems shall not be described in detail in this section – we present the Steihaug conjugate gradient method in functions space for the solution of both linear and nonlinear boundary value problems in the context of the reduced approach in Subsection 6.1.6.

6.1.5 Applicability of Newton's Method for the Interval Problems

At first sight, the application of Newton's method works quite well for the solution of the example problems considered in the introduction. However, application to more complicated problems leads to poor convergence results or no convergence at all, which we investigate in the following. For reasons of simplicity, we consider the solution of a nonlinear ODE initial value problem, the well known instable *Lorenz attractor*, by Newton's method as presented above and hereby point out the weak points of the approach.

Example 6.3. (Lorenz attractor) We consider the nonlinear system of ordinary differential equations

$$\begin{aligned} \partial_t u_1 &= -a u_1 + a u_2, & u_1(0) &= 10, \\ \partial_t u_2 &= -b u_1 - u_2 - u_1 u_3, & u_2(0) &= 0, \\ \partial_t u_3 &= -c u_3 + u_1 u_2, & u_3(0) &= 25, \end{aligned} \tag{6.15}$$

with given parameters

$$a = 10, \quad b = 28, \quad c = 2.67.$$

The *efficient solution* of the *non stiff* Lorenz attractor system is usually performed by higher order explicit time stepping schemes. However, in our exemplary context it is sufficient to consider the solution with the implicit Euler time stepping scheme. The solution of (6.15) on a time interval $I = (0, 20)$ with step size 0.002 and application of Newton's method in every time step yields the commonly known solution of the Lorenz attractor system shown in Figure 6.5.

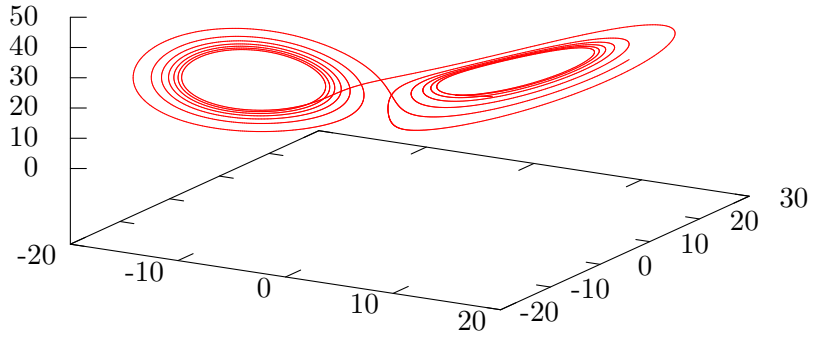


Figure 6.5: Solution of the Lorenz attractor.

Now, we apply Newton's method to the nonlinear system of equations and solve the linearized system for the Newton update $\tilde{u}$ by the same implicit Euler time stepping scheme as before.

The resulting linear system is given by the equations

$$\begin{aligned}
\partial_t \tilde{u}_1 &= -a\tilde{u}_1 + a\tilde{u}_2 - r_1, \\
\partial_t \tilde{u}_2 &= -b\tilde{u}_1 - \tilde{u}_2 - \tilde{u}_3 u_3 - u_1\tilde{u}_3 - r_2, \\
\partial_t \tilde{u}_3 &= -c\tilde{u}_3 + \tilde{u}_1 u_2 + u_1\tilde{u}_2 - r_3,
\end{aligned}$$

with homogenous initial values and residuals as follows:

$$\begin{aligned}
\tilde{u}_1(0) &= 0, & r_1 &= \partial_t u_1 - (-au_1 + au_2), \\
\tilde{u}_2(0) &= 0, & r_2 &= \partial_t u_2 - (-bu_1 - u_2 - u_1 u_3),. \\
\tilde{u}_3(0) &= 0, & r_3 &= \partial_t u_3 - (-cu_3 + u_1 u_2).
\end{aligned}$$

We observe that even on small time intervals the Newton iterates converge very slowly to the solution. Regarding Figure 6.6, we see the dependence of the number of Newton steps performed until convergence on the length of the time interval. The reason for this behavior of

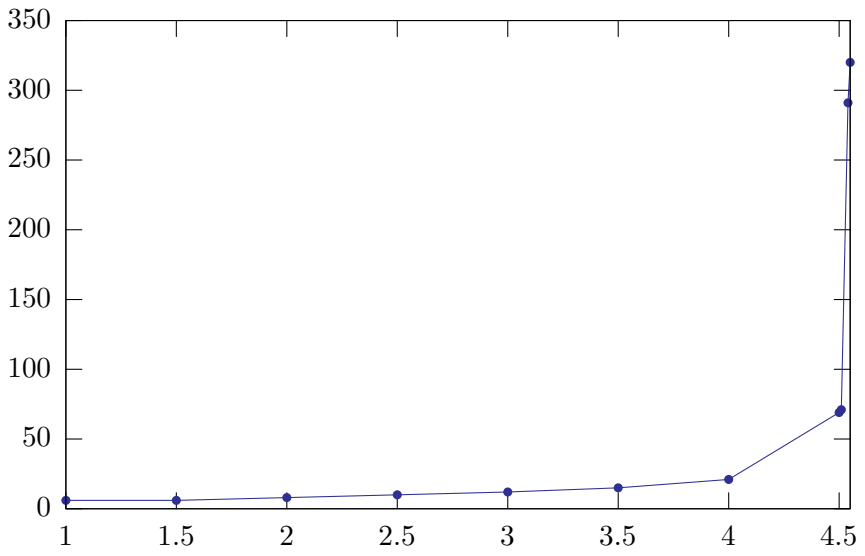
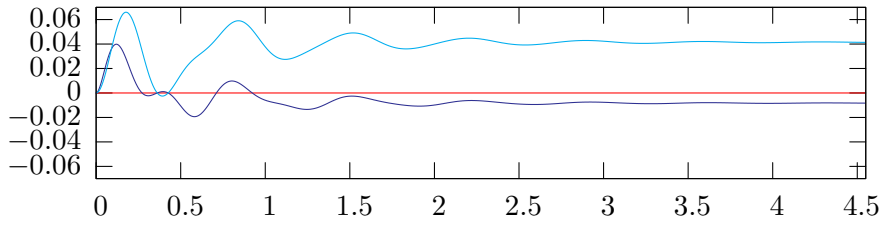


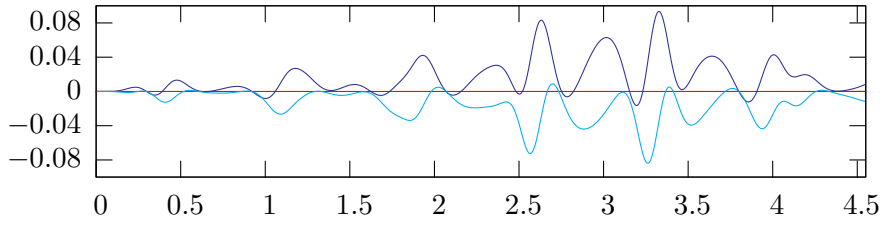
Figure 6.6: Convergence behavior of Newton's method for different intervals $I = (0, T)$. (x-axis: T, y-axis: steps until convergence)

Newton's method can be illustrated by visualizing the Newton residuals for different Newton steps. This is done in Figure 6.7 from which we clearly see that within the Newton method, the error is annihilated successively in time, but simultaneously the error at the later time steps grows exponentially.

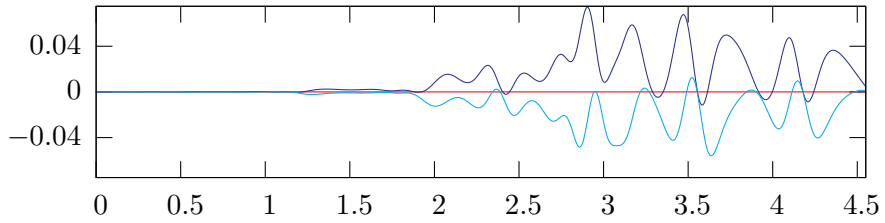
This behavior can be understood by considering the fact that the application of an implicit Euler time stepping scheme leads to a fast growing approximation error of the Newton increment in time. Therefore, the Newton updates for the solution on earlier time steps are close to the exact update while the updates at later time steps are only a poor approximation. Thus the next Newton iterate is a better approximation to the solution on early time steps but a worse in later ones. Now, in each Newton step, further time steps of the update can



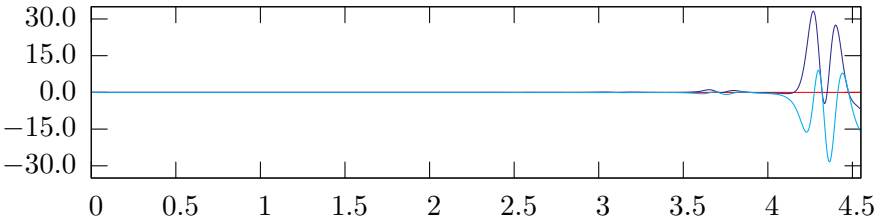
(a) $l = 2$



(b) $l = 4$



(c) $l = 6$



(d) $l = 8$

Figure 6.7: Newton residuals r_1 (red), r_2 (dark blue), r_3 (blue) with increasing number l of Newton steps from (a) to (d). (x-axis: time, y-axis: error)

be approximated quite well and the residuals vanish successively, but slowly, after a certain number of Newton steps.

On the other hand, the solution of the problem with the common approach, that is the application of the time stepping scheme to the nonlinear problem and solving each time step with Newton's method, works quite well. The reason for this different behavior of Newton's

method can be revealed by means of the condition numbers of the system matrices. It is well known that the Lorenz attractor system is locally well conditioned and globally ill conditioned as explained in the following. For the common approach, in each Newton step a system matrix of size 3×3 is inverted, whereas one Newton step of the Newton approach for the whole problem can be interpreted as the solution of a linear system of size $3 \cdot n \times 3 \cdot n$ where n denotes the number of time steps. This large system matrix is obviously built up from the time step matrices on the diagonal and additional identity matrices on the upper secondary diagonal. Figure 6.8 points out, how the condition of the system matrix changes with the enlargement of the matrix. While the small matrices of each time step have a condition of approximately 1, the condition of the large system grows in time, and so does the error. Accordingly, for fixed terminal time and decreasing time step size, the condition grows, too,

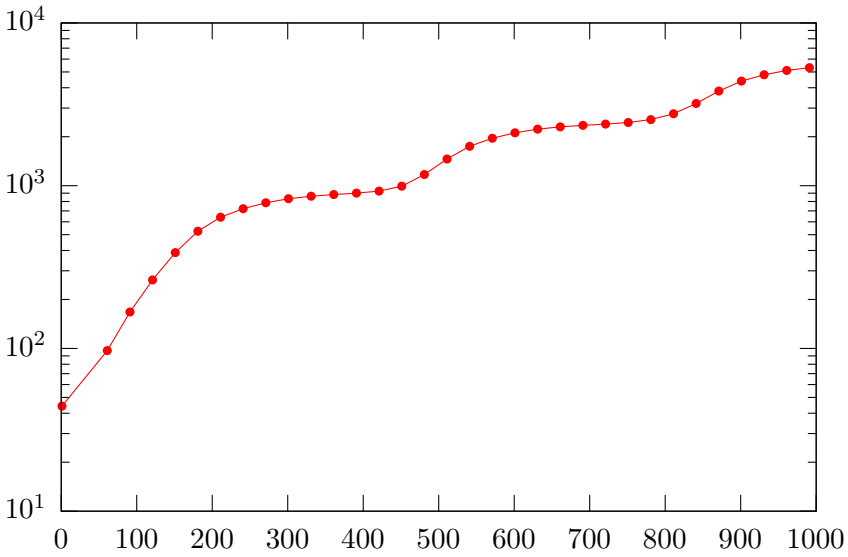


Figure 6.8: Condition number with growing T for fixed time step size. (x-axis: number of time steps, y-axis: condition)

as shown in Figure 6.9. Over all, we can summarize that Newton's method for the solution of the initial value problem has crucial drawbacks which lead to a breakdown of the method.

The problems discussed above can directly be transferred to the application on the intervalwise boundary value problems of the indirect multiple shooting approach. Thus, we do not follow this approach further on, but present an alternative method for the solution of both linear and nonlinear boundary value problems.

6.1.6 Solution of the Interval Problems – The Reduced Approach

The second approach starts with the reformulation of the nonlinear boundary value problems in terms of intervalwise optimization problems which can be solved by application of the *reduced approach*. This transformation is done by calculation of the Lagrangian by means of evaluating the antiderivative with respect to the states and control.

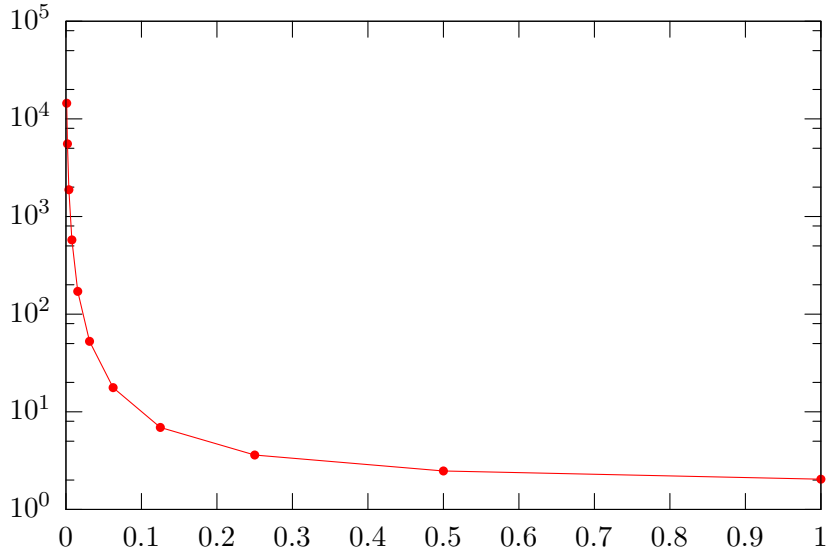


Figure 6.9: Condition number with decreasing time step size for fixed T. (x-axis: time step size, y-axis: condition)

First, we set up the optimization problem

$$\min_{(q^j,u^j)\in Q^j\times X^j} J^j(q^j,u^j) := J_1^j(u^j) + \frac{\alpha_3}{2}\|q^j\|_{Q^j}^2 + (u^j(\tau_{j+1}),\lambda^{j+1}) \tag{6.16a}$$

such that (q^j,u^j) fulfills the constraining equation

$$((\partial_t u^j,\varphi))_j + a(u^j)(\varphi) + b(q^j)(\varphi) + (u^j(\tau_j) - s^j,\varphi(\tau_j)) = ((f,\varphi))_j \quad \forall\varphi\in X^j. \tag{6.16b}$$

The corresponding Lagrangian is given as

$$\begin{aligned}\mathcal{L}^j(q^j,u^j,z^j) := {} & J_1^j(u^j) + \frac{\alpha_3}{2}\|q^j\|_{Q^j}^2 + (u^j(\tau_{j+1}),\lambda^{j+1}) \\ & - \left\{((\partial_t u^j,z^j))_j + a(u^j)(z^j) + b(q^j)(z^j) + (u^j(\tau_j) - s^j,z^j(\tau_j)) - ((f,z^j))_j\right\},\end{aligned}$$

and differentiation with respect to the control and the states yields the optimality system

$$((\partial_t u^j,\varphi))_j + a(u^j)(\varphi) + b(q^j)(\varphi) + (u^j(\tau_j) - s^j,\varphi(\tau_j)) = ((f,\varphi))_j \quad \forall\varphi\in X^j. \tag{6.17a}$$

$$-((\partial_t z^j,\psi))_j + a_u'(u^j)(\psi,z^j) + (z^j(\tau_{j+1}) - \lambda^{j+1},\psi(\tau_{j+1})) = \alpha_1 J_1^{j\prime}(u^j)(\psi) \quad \forall\psi\in X^j. \tag{6.17b}$$

$$b_q'(q^j)(\chi,z^j) - \alpha_3(q^j,\chi)_{Q^j} = 0 \quad \forall\chi\in Q^j. \tag{6.17c}$$

which is exactly the nonlinear boundary value problem (4.8).

Remark 6.9. (Boundedness of J^j) In the case that u depends linearly on q, that is $u=\gamma q$, and a cost functional as stated in Remark 2.4 it can easily be verified that the cost functional J^j in (6.16) is bounded from below. With $J_1^j(u^j) + \frac{\alpha_3}{2}\|q^j\|_{Q^j}^2$ bounded from below due to the solvability of the original problem, and thus

$$J_1^j(u^j) = \mathcal{O}(\|q\|_{Q^j}^2) \quad \text{for} \quad \|q\|_{Q^j}\to\infty,$$

we obtain

$$\begin{aligned} J^j(q^j, u^j) &= J_1^j(u^j) + \frac{\alpha_3}{2}\|q^j\|_{Q^j}^2 + \gamma(q^j, \lambda^{j+1}) \\ &= J_1^j(u^j) + \frac{\alpha_3}{2}\|q^j\|_{Q^j}^2 + \gamma(q^j, \lambda^{j+1}) \\ &= J_1^j(u^j) + \frac{1}{2}(q^j, \alpha_3 q^j + 2\gamma\lambda^{j+1}) > -\infty. \end{aligned}$$

This is due to the fact that $\frac{1}{2}(q^j, \alpha_3 q^j + 2\gamma\lambda^{j+1}) = \mathcal{O}(\|q\|_{Q^j})$ for $\|q\|_{Q^j} \to \infty$. In the general case, the boundedness can be shown analogously by more sophisticated transformations.

In order to simplify the notation, we drop the interval index j in the following presentation. A detailed description of the reduced approach is given in [29]. Applying it to the optimization problem (6.16) we introduce the solution operator $S : Q \to X$ for the primal equation and the reduced cost functional $j : Q \to \mathbb{R}$ which is defined as

$$j(q) := J(q, Sq).$$

We can now reformulate (6.16) in terms of an unconstrained optimization problem

$$\text{Minimize} \quad j(q), \quad q \in Q,$$

for which the first order optimality condition (6.18) is obtained by differentiation with respect to the control q:

$$j'(q)(\delta q) = 0 \quad \forall \delta q \in Q. \tag{6.18}$$

In order to solve (6.18) by Newton's method, differentiation of (6.18) with respect to q leads to the linear system (6.19) for the Newton updates:

$$j''(q)(\delta q, \tau q) = -j'(q)(\tau q) \quad \forall \tau q \in Q. \tag{6.19}$$

Consequently, Newton's method needs the evaluation of the first and second derivatives of j. In [29] representation formulas for these derivatives are developed and discussed in detail. The general idea is the usage of the relation

$$j(q) = \mathcal{L}(q, u, z)$$

which is valid for $u = Sq$, and the differentiation of this expression with respect to q. Exploiting the relation $u = Sq$ we obtain

$$j'(q)(\delta q) = \mathcal{L}_u'(q, u, z)(\delta u) + \mathcal{L}_q'(q, u, z)(\delta q).$$

Keeping in mind that $\mathcal{L}_u'(q, u, z)(\varphi) = 0$ for all $\varphi \in X$ yields the dual equation, the evaluation of the first derivative of the reduced cost functional can be brought down to the following two steps:

1. Calculate $z \in X$ such that z solves the dual equation (6.17b).
2. Evaluate $j'(q)(\delta q)$ according to the identity

$$\begin{aligned} j'(q)(\delta q) &= \mathcal{L}_q'(q, u, z)(\delta q) \\ &= b_q'(q)(\delta q, z) - \alpha_3(q, \delta q)_Q. \end{aligned}$$

For the second derivative, similar ideas can be applied but a more complex notational framework is needed due to the consideration of the second derivatives of $\mathcal{L}$. The second derivative of j is obtained by differentiation and application of the chain rule. We introduce the abbreviations

$$\begin{aligned}
\delta u &:= \frac{du}{dq}(\delta q), \qquad \tau u := \frac{du}{dq}(\tau q), \qquad \delta\tau u := \frac{d^2u}{dq^2}(\delta q, \tau q),\\
\delta z &:= \frac{dz}{dq}(\delta q), \qquad \tau z := \frac{dz}{dq}(\tau q), \qquad \delta\tau z := \frac{d^2z}{dq^2}(\delta q, \tau q)
\end{aligned}$$

and obtain by elementary calculus the representation of the second derivative:

$$\begin{aligned}
j''(q)(\delta q,\tau q) = \quad & \mathcal{L}''_{qq}(q,u,z)(\delta q,\tau q) + \mathcal{L}''_{qu}(q,u,z)(\delta q,\tau u) + \mathcal{L}''_{qz}(q,u,z)(\delta q,\tau z)\\
+ \quad & \mathcal{L}''_{uq}(q,u,z)(\delta u,\tau q) + \mathcal{L}''_{uu}(q,u,z)(\delta u,\tau u) + \mathcal{L}''_{uz}(q,u,z)(\delta u,\tau z)\\
+ \quad & \mathcal{L}''_{zq}(q,u,z)(\delta z,\tau q) + \mathcal{L}''_{zu}(q,u,z)(\delta z,\tau u)\\
& + \mathcal{L}'_{u}(q,u,z)(\delta\tau u) + \mathcal{L}'_{z}(q,u,z)(\delta\tau z).
\end{aligned}$$

With u and z solutions of primal and dual equation, the last two terms of the sum vanish, and we retrieve

$$\begin{aligned}
j''(q)(\delta q,\tau q) = \quad & \mathcal{L}''_{qq}(q,u,z)(\delta q,\tau q) + \mathcal{L}''_{qu}(q,u,z)(\delta q,\tau u) + \mathcal{L}''_{qz}(q,u,z)(\delta q,\tau z)\\
+ \quad & \mathcal{L}''_{uq}(q,u,z)(\delta u,\tau q) + \mathcal{L}''_{uu}(q,u,z)(\delta u,\tau u) + \mathcal{L}''_{uz}(q,u,z)(\delta u,\tau z)\\
+ \quad & \mathcal{L}''_{zq}(q,u,z)(\delta z,\tau q) + \mathcal{L}''_{zu}(q,u,z)(\delta z,\tau u).
\end{aligned}$$

In analogy to the calculation of the first derivative, we recommend that δu and δz are determined, such that

$$\mathcal{L}''_{qz}(q,u,z)(\delta q,\varphi) + \mathcal{L}''_{uz}(q,u,z)(\delta u,\varphi) = 0 \quad \forall\varphi \in X, \tag{6.20}$$

$$\mathcal{L}''_{qu}(q,u,z)(\delta q,\psi) + \mathcal{L}''_{uu}(q,u,z)(\delta u,\psi) + \mathcal{L}''_{zu}(q,u,z)(\delta z,\psi) = 0 \quad \forall\psi \in X. \tag{6.21}$$

These additional equations (6.20) and (6.21) are denoted as the *tangent equation* and the *additional adjoint equation*. Reattaching the interval index, these equations read

$$((\partial_t \delta u^j, \varphi))_j + a'_u(u^j)(\delta u,\varphi) + b'_q(q^j)(\delta q^j,\varphi) + (\delta u^j(\tau_j),\varphi(\tau_j)) = 0 \quad \forall\varphi \in X^j, \tag{6.22}$$

$$\begin{aligned}
-((\partial_t \delta z^j, \psi))_j + a'_u(u^j)(\psi,\delta z^j) + (\delta z^j(\tau_{j+1}),\psi(\tau_{j+1})) + a''_{uu}(u^j)(\delta u^j,\psi,\delta z^j)\\
- \alpha_1 J_1^{j\prime\prime}(u^j)(\delta u^j\psi) = 0 \quad \forall\psi \in X^j.
\end{aligned} \tag{6.23}$$

The second derivative of j can now be calculated by the following procedure:

1. Solve the tangent equation (6.22) for δu.
2. Solve the additional adjoint equation (6.23) for δz .
3. Evaluate the second derivatives by means of

$$\begin{aligned}
j''(q)(\delta q,\tau q) &= \mathcal{L}''_{qq}(q,u,z)(\delta q,\tau q) + \mathcal{L}''_{uq}(q,u,z)(\delta u,\tau q) + \mathcal{L}''_{zq}(q,u,z)(\delta z,\tau q)\\
&= b'_q(q)(\delta q,\delta z) + b''_{qq}(q)(\delta q,\tau q,z) - \alpha_3(\delta q,\tau q)_Q.
\end{aligned}$$

We do not go further into detail concerning the derivation of the additional equations. A thorough investigation and description can be found in the literature mentioned in the introduction of this paragraph. Instead, we focus on the setup of a globalized Newton method for the solution of (6.18). So far in this subsection, we have considered the undiscretized, infinite dimensional problem formulation, while in the following, when describing the Newton method, we want consider again the finite dimensional case. Therefore, we need the discrete solution operator $S_{kh} : Q_d \to X_{kh}^{j,r,s}$, according to the discretization of (6.17a), and furthermore all partial differential equations (6.17b), (6.22), (6.23) have to be replaced by their discretizations. Consequently, from now on, we consider the reduced functional

$$j_{kh}(q_\sigma) := J(q_\sigma, S_{kh}(u_\sigma)).$$

Remark 6.10. For the ease of presentation, Newton's method for this problem is presented in function space. Whenever necessary, we hint at the formulation in terms of the coefficient vectors.

We have already discussed, that one step of Newton's method requires the solution of the linear system (6.19), which is the first order optimality condition of the linear quadratic optimal control problem

$$\min_{\delta q_\sigma} m(q_\sigma, \delta q_\sigma) := j_{kh}(q_\sigma) + j'_{kh}(q_\sigma)(\delta q_\sigma) + \frac{1}{2} j''_{kh}(q_\sigma)(\delta q_\sigma, \delta q_\sigma). \tag{6.24}$$

Thus, if δq_σ is a solution of (6.24), it is a solution of (6.19). Furthermore, if the second derivative is positive definite, $j''_{kh}(q_\sigma)(\delta q_\sigma, \delta q_\sigma) > 0 \quad \forall \delta q_\sigma \in Q_d^{\neq 0}$, the first order optimality condition is not only necessary but also sufficient, and the reversal holds, too. In the following, we demand an additional constraint on the Newton update, $\|\delta q_\sigma\|_Q \leq \mu$, $\mu \in \mathbb{R} \cup \{\infty\}$, for this problem, which allows us to apply a conjugate gradient method in combination with trust region globalization techniques:

$$\min_{\delta q_\sigma} m(q_\sigma, \delta q_\sigma) := j(q_\sigma) + j'(q_\sigma)(\delta q_\sigma) + \frac{1}{2} j''_{kh}(q_\sigma)(\delta q_\sigma, \delta q_\sigma) \quad \text{s.t.} \quad \|\delta q_\sigma\|_Q \leq \mu. \tag{6.25}$$

Remark 6.11. (Interpretation of gradient and Hessian) The formulation of Newton's method in function space requires the setup and calculation of the gradient $\nabla j_{kh}(q_\sigma) \in Q_d$ and the Hessian $\nabla^2 j_{kh}(q_\sigma) : Q_d \to Q_d$ which have to be understood by means of the Hilbert space identifications

$$\begin{aligned} (\nabla j(q_\sigma), \tau q)_Q &= j'_{kh}(q_\sigma)(\tau q) \quad \forall \tau q \in Q_d, \\ (\nabla^2 j_{kh}(q_\sigma)(\delta q_\sigma), \tau q)_Q &= (j'_{kh}(q_\sigma)(\delta q_\sigma), \tau q)_Q \quad \forall \tau q \in Q_d. \end{aligned}$$

Remark 6.12. (Calculation of discrete gradient and Hessian) In the discrete, finite dimensional case, we have to consider the vector representation with respect to the basis of the discretized control space. The coefficient vectors representing gradient and Hessian can be calculated explicitly. Exemplary with $\{ \tau q_0, \ldots, \tau q_n \}$ a basis of the discrete control space Q_d, the vector g for the gradient is obtained as the solution of the linear system

$$Gg = r$$

for which the Gramian matrix G and right-hand side r are determined via

$$G_{ij} = (\tau q_i, \tau q_j)_Q \quad \text{and} \quad r_i = j'_{kh}(q_\sigma)(\tau q_i).$$

Accordingly, the vector representation d of the directional derivative $\nabla^2 j_{kh}(q_\sigma)(\delta q_\sigma)$ is obtained from the linear system

$$Gd = h$$

with right-hand side

$$h_i = j''_{kh}(q_\sigma)(\delta q_\sigma, \tau q_i).$$

By means of these concretizations, all algorithms presented in this subsection can easily be brought forward to the vector based implementation of the approach.

With the equivalence of (6.24) and (6.19) at hand, we can formulate the Newton algorithm in function space.

Algorithm 6.6 Newton's method for the solution of (6.18)

1: Start with an initial guess $q_{\sigma,0} \in Q_d$, $\mu_0 \in \mathbb{R} \cup \{+\infty\}$, set $l = 0$.
2: **repeat**
3: Solve the state equation for $u_{\sigma,l}$.
4: Solve the dual equation for $z_{\sigma,l}$.
5: Calculate the gradient $\nabla j_{kh}(q_{\sigma,l})$ according to Remark 6.11.
6: Solve the problem

$$\min_{\delta q_\sigma \in Q_d} m(q_{\sigma,l}, \delta q_\sigma) \quad \text{s.t.} \quad \|\delta q_\sigma\|_Q < \mu_l.$$

7: Choose μ_{l+1} and ν_l according to the behavior of the algorithm.
8: Calculate next iterate $q_{\sigma,l+1} = q_{\sigma,l} + \nu_l \delta q_\sigma$.
9: Set $l = l + 1$.
10: **until** $\|\nabla j_{kh}(q_{\sigma,l})\|_Q < \text{tol}$

The solution of the constrained minimization problem in step 6 of Algorithm 6.6 is preferentially done by use of an iterative solver which requires merely the evaluation of $\nabla^2 j(q_{\sigma,l})(\delta q_\sigma)$. The evaluation procedure for this second derivative can be performed as described in Algorithm 6.7.

Algorithm 6.7 Calculation of $\nabla^2 j_{kh}(q_{\sigma,l})(\delta q_\sigma)$

Require: $u_{\sigma,l}$ and $z_{\sigma,l}$ have already been computed for given $q_{\sigma,l}$.
1: Solve the discretized tangent equation for $\delta u_{\sigma,l}$.
2: Solve the discretized additional adjoint equation for $\delta z_{\sigma,l}$.
3: Obtain $\nabla^2 j(q_\sigma)(\delta q_\sigma)$ according to Remark 6.11.

Furthermore, the region of interest in each step is determined by the value of the radius μ_l, and the actual Newton step is specified by the choice of the damping parameter ν_l.

Both values are determined anew in each step of Newton's method, in order to optimize the convergence behavior. These globalization techniques are described at the end of this subsection. Next, we consider the solution of the linear problem (6.25) for the evaluation of the Newton update δq_σ which was not yet determined for step 6 of Algorithm 6.6. We therefor apply the Steihaug conjugate gradient method as described in the sequel. The method for the solution of the constrained optimization problem was developed in [38], and the algorithm for our function space setting is recapitulated in Algorithm 6.8. This algorithm terminates,

1. If the curvature of the calculated direction is negative. In this case we move to the boundary of the domain if its radius is finite, $\mu_l < \infty$, otherwise we take the previous iterate.
2. If the norm of the iterate is too large. Then we take a linear combination of the previous iterate and the current one which lies on the boundary of the domain.
3. If the Newton update is approximated sufficiently well.

Algorithm 6.8 The Steihaug conjugate gradient method for the solution of (6.25)

Set $p_0 = 0$, $r_0 = -\nabla j'_{kh}(q_\sigma)$, $g_0 = r_0$ and $i = 0$.
loop
  Compute the directional derivative $h = \nabla^2(q_{\sigma,l})(g_i)$ using Algorithm 6.7.
  Set $\gamma = (h, g_i)_Q$.
  if $\gamma \leq 0$ **then**
    if $\mu_l < \infty$ **then**
      Compute $\xi > 0$ such that $\|p_i + \xi g_i\|_Q = \mu_l$.
      Set $\delta q_\sigma = p_i + \xi g_i$.
    else
      Set $\delta q_\sigma = p_{i-1}$ or $d = p_0$ if $i = 0$.
    break *(Negative curvature found.)*
  Compute $\alpha = {|r_i|^2}/{\gamma}$.
  Set $p_{i+1} = p_i + \alpha g_i$.
  if $\|p_{i+1}\|_Q > \mu_l$ **then**
    Compute $\xi > 0$ such that $\|p_i + \xi g_i\|_Q = \mu_l$.
    Set $\delta q = p_i + \xi g_i$.
    break *(Norm of approximation too large.)*
  Compute $r_{i+1} = r_i - \alpha h$.
  if ${\|r_{i+1}\|_Q}/{\|r_0\|_Q} < \mathrm{tol}$ **then**
    Set $\delta q_\sigma = p_{i+1}$
    break *(Approximation good enough.)*
  Compute $\beta = {\|r_{i+1}\|_Q^2}/{\|r_i\|_Q^2}$.
  Set $g_{i+1} = r_{i+1} + \beta g_i$.
  Set $i = i + 1$.

Finally, we have to specify the choice of μ and ν in the context of globalization techniques of Newton's method. We are faced with the choice between line search methods and trust region methods. In this context, we have decided to apply a trust region globalization.

Algorithm 6.9 Determination of μ_{l+1} and ν_l

if $\rho_l < 0.25$ **then**
    Set $\mu_{l+1} = 0.25\|\delta q_\sigma\|_Q$.
else if $\rho_l > 0.75$ and $\|\delta q_\sigma\|_Q = \mu_l$ **then**
    Set $\mu_{l+1} = \min(2\mu_l, \mu_{\max})$.
else
    Set $\mu_{l+1} = \mu_l$.
if $\rho_l > \theta$ **then**
    Set $\nu_l = 1$.
else
    Set $\nu_l = 0$.

A Newton step is either accepted or rejected, that is $\nu_l = 1$ or $\nu_l = 0$ while the diameter $\mu_l < \infty$ of the current trust region is adapted in each Newton step. According to the textbook of Nocedal and Wright [32] the trust region method is determined by the choice of its parameters, which are the maximal radius $\mu_{\max} > 0$, the initial radius $\mu_0 \in (0, \mu_{\max})$ and a parameter $\theta \in [0, 0.25)$ by which we determine the minimum performance needed for the Newton step to be accepted. Therefore, we define the ratio

$$\rho_l := \frac{j_{kh}(q_{\sigma,l}) - j_{kh}(q_{\sigma,l} + \delta q_\sigma)}{m(q_{\sigma,l}, 0) - m(q_{\sigma,l}, \delta q_\sigma)}$$

which gives information on the quality of the approximation of j by the functional m. The parameters μ_{l+1} and ν_l are then determined by Algorithm 6.9.

So far, we have seen how the (nonlinear) intervalwise boundary value problems can be solved by the reduced approach for PDE constrained optimization problems. We have shortly presented the basic ideas of this approach and pointed out the algorithmic aspects of the solvers. Next, the solution techniques for the direct multiple shooting approach are presented analogously.

6.2 Solution Techniques for the Direct Multiple Shooting Approach

The solution techniques for the direct approach start analogously to the indirect approach with the application of *Newton's method* to the system of matching conditions. The linearized system is either solved directly by application of a *Krylov subspace method*, similar to the approach in [14], or otherwise a *condensing technique* reduces the problem to a linear system on the control space. The condensed direct multiple shooting approach comes close to the multiple shooting approach for ODE constrained optimization. Therefore, we point out at the end of this section why certain efficient solution techniques from the ODE approach can not be transferred to the PDE approach, and consequently why the PDE approach has limitations with respect to the efficiency.

Newton's method requires the evaluation of the residual, and the linear solver needs the calculation of the directional derivatives forming the left-hand side of the linearized problem. Therefore, the linear solver requires the solution of intervalwise linear initial value problems (IVPs) while the calculation of the residual needs the solution of nonlinear initial value problems. The interrelation of the different problems is shown in Figure 6.10.

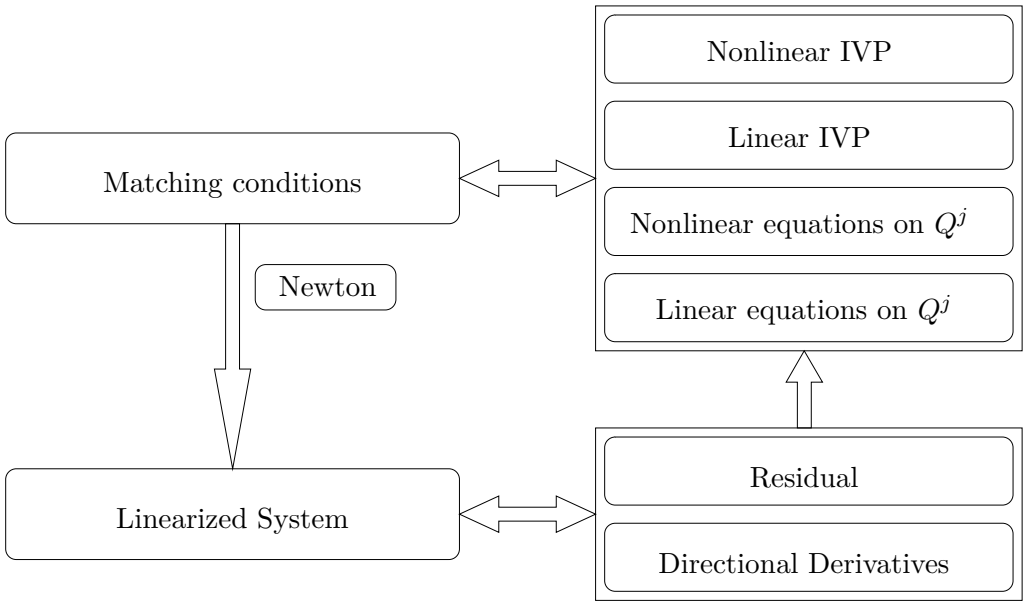


Figure 6.10: Different solution approaches for direct multiple shooting.

6.2.1 Solution of the Multiple Shooting System

The linear system determining the Newton update $(\Delta\tilde{q}_\sigma, \Delta\tilde{s}_h, \Delta\tilde{p}_h)$ is obtained by differentiation of (4.35) with respect to the multiple shooting variables.

Remark 6.13. In the following, we consider s^0 and p^m as defined by the corresponding equations in (4.35). Therefore, these variables do not have to be determined by Newton's method, and we can define $\Delta s^0 := 0$ and $\Delta p^m := 0$.

We obtain for $j = 0, \ldots, m-1$, the following equations:

$$(\Delta s_h^{j+1} - \bar{S}'_{js^j}(\Delta s_h^j) - \bar{S}'_{jq^j}(\Delta q_\sigma^j), \varphi) = -(r_u^j, v) \qquad \forall \varphi \in H_h^{j+1},$$

$$(\Delta p_h^j - \bar{\Xi}'_{jp^{j+1}}(\Delta p_h^{j+1}) - \bar{\Xi}'_{js^j}(\Delta s_h^j) - \bar{\Xi}'_{jq^j}(\Delta q_\sigma^j), \psi) = -(r_z^j, \psi) \qquad \forall \psi \in H_h^j,$$

$$\begin{aligned} &\alpha_3(\Delta q_\sigma^j, \chi)_{Q^j} - b''_{qq}(q^j)(\Delta q_\sigma^j, \chi, \Xi_j(p_h^{j+1}, s_h^j, q_\sigma^j)) \\ &\quad -b'_q(q_\sigma^j)(\chi, \Xi'_{js^j}(\Delta s_h^j) + \Xi'_{jq^j}(\Delta q_\sigma^j) + \Xi'_{jp^{j+1}}(\Delta p_h^{j+1})) = -(r_q^j, \chi)_{Q^j} \quad \forall \chi \in Q_d^j \end{aligned} \tag{6.26}$$

with the Newton residuals defined as

$$\begin{aligned}(r_u^j, \varphi) &:= (s_h^{j+1} - \bar{S}_j(s_h^j, q_\sigma^j), \varphi),\\ (r_z^j, \psi) &:= (p_h^j - \bar{\Xi}_j(p_h^{j+1}, s_h^j, q_\sigma^j), \psi),\\ (r_q^j, \chi)_{Q^j} &:= \alpha_3(q_\sigma^j, \chi)_{Q^j} - b(q_\sigma^j)(\chi, \Xi_j(p_h^{j+1}, s_h^j, q_\sigma^j)).\end{aligned}$$

Here, the derivatives of the solution operators for the primal equation have to be understood according to Lemma 4.3. For the solution operators of the dual equation an analogous statement holds and can easily be derived by differentiation of the dual equation with respect to p^{j+1}, s^j, and q^j. We skip the explicit formulation but annotate that the discrete solution operators as considered here are defined accordingly for the discrete equations. We continue with the introduction of an abbreviatory notation as done in (6.2) by introducing additional operators E_j for $j = 1, \dots, m-1$, F_j for $j = 0, \dots, m-1$, and G_j for $j = 0, \dots, m-2$ in weak formulation:

$$\begin{aligned}(E_j(\Delta s_h^j), \chi) &:= b_q'(q_\sigma^j)(\chi, \Xi_{js^j}'(\Delta s_h^j)),\\ (F_j(\Delta q_\sigma^j), \chi) &:= \alpha_3(\Delta q_\sigma^j, \chi)_{Q^j} - b_{qq}''(q_\sigma^j)(\Delta q_\sigma^j, \chi, \Xi_j(p_h^{j+1}, s_h^j, q_\sigma^j)) - b_q'(q_\sigma^j)(\chi, \Xi_{jq_\sigma^j}'(\Delta q_\sigma^j)),\\ (G_j(\Delta p_h^{j+1}), \chi) &:= b_q'(q_\sigma^j)(\chi, \Xi_{jp^{j+1}}'(\Delta p_h^{j+1})).\end{aligned}$$

We define the pseudo matrices for the different blocks of the linearized system:

$$A_0 := \begin{pmatrix} I & -\bar{S}_{0,q^0}' \\ 0 & F_0 \end{pmatrix}, \quad A_j := \begin{pmatrix} I & -\bar{S}_{j,q^j}' & 0 \\ 0 & F_j & 0 \\ 0 & -\bar{\Xi}_{j,q^j}' & I \end{pmatrix}, \qquad A_{m-1} := \begin{pmatrix} F_j & 0 \\ -\bar{\Xi}_{m-1,q^{m-1}}' & I \end{pmatrix},$$

$$B_0 := \begin{pmatrix} -\bar{S}_{1,s^1}' & 0 \\ -E_1 & 0 \\ -\bar{\Xi}_{1,s^1}' & 0 \end{pmatrix}, \quad B_j := \begin{pmatrix} -\bar{S}_{j+1,s^{j+1}}' & 0 & 0 \\ -E_{j+1} & 0 & 0 \\ -\bar{\Xi}_{j+1,s^{j+1}}' & 0 & 0 \end{pmatrix}, \quad B_{m-2} := \begin{pmatrix} -E_{m-1} & 0 & 0 \\ -\bar{\Xi}_{m-1,s^{m-1}}' & 0 & 0 \end{pmatrix},$$

$$C_0 := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -G_0 \end{pmatrix}, \quad C_j := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -G_j \\ 0 & 0 & -\bar{\Xi}_{jp^{j+1}}' \end{pmatrix}, \qquad C_{m-2} := \begin{pmatrix} 0 & 0 \\ 0 & -G_{m-2} \\ 0 & -\bar{\Xi}_{m-2p^{m-1}}' \end{pmatrix}.$$

Furthermore, for the right-hand side and the increment, we define for the different intervals

$$\begin{aligned}\begin{pmatrix} (r_{0,0}, \varphi) \\ (r_{0,1}, \chi)_{Q^j} \end{pmatrix} &:= \begin{pmatrix} (s_h^1 - \bar{S}_0(s_h^0, q_\sigma^j), \varphi) \\ \alpha_3(q_\sigma^0, \chi)_{Q^j} - b(q_\sigma^j)(\chi, \Xi_j(p_h^1, s_h^0, q_\sigma^0)) \end{pmatrix},\\ \begin{pmatrix} (r_{j,0}, \varphi) \\ (r_{j,1}, \chi)_{Q^j} \\ (r_{j,2}, \psi) \end{pmatrix} &:= \begin{pmatrix} (s_h^{j+1} - \bar{S}_j(s_h^j, q_\sigma^j), \varphi) \\ \alpha_3(q_\sigma^j, \chi)_{Q^j} - b(q_\sigma^j)(\chi, \Xi_j(p_h^{j+1}, s_h^j, q_\sigma^j)) \\ (p_h^j - \bar{\Xi}_j(p_h^{j+1}, s_h^j, q_\sigma^j), \psi) \end{pmatrix},\\ \begin{pmatrix} (r_{m-1,0}, \chi)_{Q^j} \\ (r_{m-1,1}, \psi) \end{pmatrix} &:= \begin{pmatrix} \alpha_3(q_\sigma^{m-1}, \chi)_{Q^{m-1}} - b(q_\sigma^{m-1})(\chi, \Xi_j(p_h^m, s_h^{m-1}, q_\sigma^{m-1})) \\ (p_h^{m-1} - \bar{\Xi}_j(p_h^m, s_h^{m-1}, q_\sigma^{m-1}), \psi) \end{pmatrix},\\ x_0 &:= (\Delta s_h^1, \Delta q_\sigma^0)^T,\\ x_j &:= (\Delta s_h^{j+1}, \Delta q_\sigma^j, \Delta p_h^j)^T,\\ x_{m-1} &:= (\Delta q_\sigma^{m-1}, \Delta p_h^{m-1})^T,\end{aligned}$$

such that the linear system (6.26) reads in pseudo matrix notation

$$\underbrace{\begin{pmatrix} A_0 & C_0 & & & \\ B_0 & A_1 & C_1 & & \\ & \ddots & \ddots & \ddots & \\ & & B_{m-3} & A_{m-2} & C_{m-2} \\ & & & B_{m-2} & A_{m-1} \end{pmatrix}}_{=:K} \underbrace{\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{m-2} \\ x_{m-1} \end{pmatrix}}_{=:x} = - \underbrace{\begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{m-2} \\ r_{m-1} \end{pmatrix}}_{=:b}. \tag{6.28}$$

The algorithmic formulation of Newton's method for the solution of the direct multiple shooting approach is equivalent to the formulation for the indirect approach except for the definition of the linear system. The explicit formulation is given in Algorithm 6.10.

Algorithm 6.10 Newton's method for the direct approach

1: Set $i = 0$.
2: Start with initial guess for multiple shooting variables

$$x_i = (s^1_{h,i}, q^0_\sigma, s^2_{h,i}, q^1_\sigma, p^1_h, \ldots, s^{m-1}_{h,i}, q^{m-2}_{\sigma,i}, p^{m-2}_{h,i}, q^{m-1}_{\sigma,i}, p^{m-1}_{h,i})^T.$$

3: **repeat**
4: Calculate the residuals $b_i = (r_{0,i}, \ldots, r_{m-1,i})^T$ according to (6.3).
5: Solve the linear system (6.28) for the Newton update

$$K_i \Delta x_i = -b_i,$$

$$\Delta x_i = (\Delta s^1_{h,i}, \Delta q^0_\sigma, \Delta s^2_{h,i}, \Delta q^1_\sigma, \Delta p^1_h, \ldots, \Delta s^{m-1}_{h,i}, \Delta q^{m-2}_{\sigma,i}, \Delta p^{m-2}_{h,i}, \Delta q^{m-1}_{\sigma,i}, \Delta p^{m-1}_{h,i})^T.$$

6: Calculate next iterate

$$x_{i+1} = x_i + \Delta x_i.$$

7: Set $i = i + 1$.
8: **until** $\|b_i\| := \sqrt{\|r_{0,i}\|^2_{H^1_h \times Q^0_d} + \sum_{j=0}^{m-1} \|r_{j,i}\|^2_{H^{j+1}_h \times Q^j_d \times H^j_h} + \|r_{m-1,i}\|^2_{Q^{m-1}_d \times H^{m-1}_h}} < \text{tol}.$

Remark 6.14. (Matrix-vector based formulation of Newton's method) The consideration of Newton's method in the discrete case needs the determination of the coefficient vectors of the solutions from a linear system of equations. Therefore, we need the concretization of the linear system (6.28) for given bases of the discretized spaces. Again, we do this exemplarily for only one block row $(B_{j-1} \quad A_j \quad C_j)$ of the pseudo block matrix K. We memorize the corresponding equations of the linearized system (6.26).

$$(\Delta s^{j+1}_h - \bar{S}'_{js^j}(\Delta s^j_h) - \bar{S}'_{jq^j}(\Delta q^j_\sigma), \varphi) = -(r^j_u, \varphi) \qquad \forall \varphi \in H^{j+1}_h,$$

$$(\Delta p^j_h - \bar{\Xi}'_{jp^{j+1}}(\Delta p^{j+1}_h) - \bar{\Xi}'_{js^j}(\Delta s^j_h) - \bar{\Xi}'_{jq^j}(\Delta q^j_\sigma), \psi) = -(r^j_z, \psi) \qquad \forall \psi \in H^j_h,$$

$$\begin{aligned} &\alpha_3(\Delta q^j_\sigma, \chi)_{Q^j} - b''_{qq}(q^j)(\Delta q^j_\sigma, \chi, \Xi_j(p^{j+1}_h, s^j_h, q^j_\sigma)) \\ &\quad -b'_q(q^j_\sigma)(\chi, \Xi'_{js^j}(\Delta s^j_h) + \Xi'_{jq^j}(\Delta q^j_\sigma) + \Xi'_{jp^{j+1}}(\Delta p^{j+1}_h)) = -(r^j_q, \chi)_{Q^j} \quad \forall \chi \in Q^j_d. \end{aligned}$$

We assume that $\varphi_0^j, \ldots, \varphi_{n_j}^j$ is a basis of the discrete space H_h^j, and that $\chi_0^j, \ldots, \chi_{\nu_j}^j$ is a basis of the discrete control space Q_d^j. We denote the coefficient vectors for the Newton updates Δs_h^j, Δp_h^j and Δq_σ^j with respect to these bases by $\Delta\xi^{j,s} \in \mathbb{R}^{n_j}$, $\Delta\xi^{j,p} \in \mathbb{R}^{n_j}$, and $\Delta\xi^{j,q} \in \mathbb{R}^{\nu_j}$. With this notation at hand, we are able to rewrite the linear equations as

$$\begin{aligned}
\mathcal{I}_0\Delta\xi^{j+1,s} - \mathcal{B}_0\Delta\xi^{j,s} - \mathcal{A}_0\Delta\xi^{j,q} &= -d_0,\\
\mathcal{I}_1\Delta\xi^{j,p} - \mathcal{C}_1\Delta\xi^{j+1,p} - \mathcal{B}_1\Delta\xi^{j,s} - \mathcal{A}_1\Delta\xi^{j,q} &= -d_1,\\
-\mathcal{C}_2\Delta\xi^{j+1,p} - \mathcal{B}_2\Delta\xi^{j,s} - \mathcal{A}_2\Delta\xi^{j,q} &= -d_2.
\end{aligned}$$

The matrices $\mathcal{I}_0 \in \mathbb{R}^{n_{j+1}\times n_{j+1}}$, $\mathcal{I}_1 \in \mathbb{R}^{n_j\times n_j}$, $\mathcal{A}_0 \in \mathbb{R}^{n_{j+1}\times \nu_j}$, $\mathcal{A}_1 \in \mathbb{R}^{n_j\times \nu_j}$, $\mathcal{A}_2 \in \mathbb{R}^{\nu_j\times \nu_j}$, $\mathcal{B}_0 \in \mathbb{R}^{n_{j+1}\times n_j}$, $\mathcal{B}_1 \in \mathbb{R}^{n_j\times n_j}$, $\mathcal{B}_2 \in \mathbb{R}^{\nu_j\times n_j}$, $\mathcal{C}_1 \in \mathbb{R}^{n_j\times n_{j+1}}$, $\mathcal{C}_2 \in \mathbb{R}^{\nu_j\times n_{j+1}}$ have to be understood according to the following identities:

$$\begin{aligned}
(\mathcal{I}_0)_{li} &= (\varphi_i^{j+1}, \varphi_l^{j+1}),\\
(\mathcal{I}_1)_{li} &= (\varphi_i^{j}, \varphi_l^{j}),\\
(\mathcal{A}_0)_{li} &= (\bar{S}'_{jq^j}(\chi_i^j), \varphi_l^{j+1}),\\
(\mathcal{A}_1)_{li} &= (\bar{\Xi}'_{jq^j}(\chi_i^j), \varphi_l^{j}),\\
(\mathcal{A}_2)_{li} &= -\alpha_3(\chi_i^j, \chi_l^j)_{Q^j} + b''_{qq}(q^j)(\chi_i^j, \chi_l^j, \Xi_j(p_h^{j+1}, s_h^j, q_\sigma^j)) + b'_q(q_\sigma^j)(\chi_l^j, \Xi'_{jq^j}(\chi_i^j)),\\
(\mathcal{B}_0)_{li} &= (\bar{S}'_{js^j}(\varphi_i^j), \varphi_l^{j+1}),\\
(\mathcal{B}_1)_{li} &= (\bar{\Xi}'_{js^j}(\varphi_i^j), \varphi_l^{j}),\\
(\mathcal{B}_2)_{li} &= b'_q(q_\sigma^j)(\chi_l^j, \Xi'_{js^j}(\varphi_i^j)),\\
(\mathcal{C}_1)_{li} &= (\bar{\Xi}'_{jp^{j+1}}(\varphi_i^{j+1}), \varphi_l^{j}),\\
(\mathcal{C}_2)_{li} &= b'_q(q_\sigma^j)(\chi_l^j, \Xi'_{jp^{j+1}}(\varphi_i^{j+1})).
\end{aligned}$$

The right-hand side vectors $d_0 \in \mathbb{R}^{n_{j+1}}$, $d_1 \in \mathbb{R}^{n_j}$, and $d_2 \in \mathbb{R}^{\nu_j}$ are given according to

$$(d_0)_l = (r_u^j, \varphi_l^{j+1}), \quad (d_1)_l = (r_z^j, \varphi_l^j), \quad \text{and} \quad (d_2)_l = (r_q^j, \chi_l^j)_{Q^j}.$$

With this interpretation at hand, the transfer of the function space method to a vector based one is straightforward by setting up the discrete linear system for the determination of the coefficient vectors of the Newton updates. In addition, it is often convenient and suitable to replace the Hilbert space norm on H by the l^2-norm on the finite dimensional space $\mathbb{R}^{n_j}$.

In the following, we consider Example (2.3) to illustrate the application of Newton's method in a concrete case.

Example 6.4. (Newton's method for Example 2.3) We reconsider the multiple shooting approach for Example 2.3 with 2 intervals (Example 4.2). We remark, that s_0 and p_2 are determined by the boundary values of the original problem and are not part of the unknowns of the linearized problem:

$$\begin{aligned}
(\bar{S}'_{0,q^0}(\Delta q_\sigma^0) - \Delta s_h^1, \varphi) &= -(\bar{S}_0(s_h^0, q_\sigma^0) - s^1, \varphi) && \forall \varphi \in H_h^1,\\
(\bar{\Xi}'_{1,s^1}(\Delta s_h^1) + \bar{\Xi}'_{1,q^1}(\Delta q_\sigma^1) - \Delta p_h^1, \psi) &= -(\bar{\Xi}_1(s_h^1, q_\sigma^1, p_h^2) - p_h^1, \psi) && \forall \psi \in H_h^1,\\
(\Delta q_\sigma^0, \chi)_{Q^0} + (\Xi'_{0,q^0}(\Delta q_\sigma^0) + \Xi'_{0,p^1}(\Delta p_h^1), \chi) &= -((q_\sigma^0, \chi)_{Q^0} + (\Xi_0(s_h^0, q_\sigma^0, p_h^1), \chi)) && \forall \chi \in Q_d^0,\\
(\Delta q_\sigma^1, \chi)_{Q^1} + (\Xi'_{1,s^1}(\Delta s_h^1) + \Xi'_{1,q^1}(\Delta q_\sigma^1), \chi) &= -((q_\sigma^1, \chi)_{Q^1} + (\Xi_1(s_h^1, q_\sigma^1, p_h^2), \chi)) && \forall \chi \in Q_d^1.
\end{aligned}$$

In pseudo matrix notation this special system is written as

$$\begin{pmatrix} -I & \bar{S}'_{0,q^0} & 0 & 0 \\ 0 & I + \Xi'_{0,q^0} & 0 & \Xi'_{0,p^1} \\ \Xi'_{1,s^1} & 0 & I + \Xi'_{1,q^1} & 0 \\ 0 & 0 & \bar{\Xi}'_{1,q^1} & -I \end{pmatrix} \begin{pmatrix} \Delta s_h^1 \\ \Delta q_\sigma^0 \\ \Delta q_\sigma^1 \\ \Delta p_h^1 \end{pmatrix} = - \begin{pmatrix} \bar{S}_0(s_h^0, q_\sigma^0) - s_h^1 \\ q_\sigma^0 + \Xi_0(s_h^0, q_\sigma^0, p_h^1) \\ q_\sigma^1 + \Xi_1(s_h^1, q_\sigma^1, p_h^2) \\ \bar{\Xi}_1(s_h^1, q_\sigma^1, p_h^2) - p_h^1 \end{pmatrix}.$$

Differentiation of the dual equation yields the determining equation for the derivatives of the solution operators, and accordingly for their discretizations by considering the corresponding discrete equations. However, we present for the sake of brevity of the equations only the continuous case.

We determine $x, y, w \in X^j$, such that for all $\varphi \in X^j$, $j = 0, 1$, the following PDEs are fulfilled:

$$\begin{aligned} &- (\!(\partial_t x^j, \varphi)\!)_j + (\!(\nabla x^j, \nabla\varphi)\!)_j + (\!(3(u^j)^2 x^j, \varphi)\!)_j + (x^j(\tau_{j+1}), \varphi(\tau_{j+1})) \\ &\qquad = (\!(S'_{j,s^j}(\xi), \varphi)\!)_j - (\!(6u^j S'_{j,s^j}(\xi) x^j, \varphi)\!)_j, \\ &- (\!(\partial_t y^j, \varphi)\!)_j + (\!(\nabla y^j, \nabla\varphi)\!)_j + (\!(3(u^j)^2 y^j, \varphi)\!)_j + (y^j(\tau_{j+1}), \varphi(\tau_{j+1})) \\ &\qquad = (\!(S'_{j,q^j}(\rho), \varphi)\!)_j - (\!(6u^j S'_{j,q^j}(\rho) x^j, \varphi)\!)_j, \\ &- (\!(\partial_t w^j, \varphi)\!)_j + (\!(\nabla w^j, \nabla\varphi)\!)_j + (\!(3(u^j)^2 w^j, \varphi)\!)_j + (w^j(\tau_{j+1}) - \zeta, \varphi(\tau_{j+1})) = 0. \end{aligned}$$

The derivatives of the solution operators with respect to s^j, q^j, and p^{j+1} are given as the solution operators of the equations above by the following mappings:

$$\begin{aligned} &\Xi'_{j,s^j}(\xi) = x, && \Xi'_{j,q^j}(\rho) = y, && \Xi'_{j,p^{j+1}}(\xi) = w, \\ &\bar{\Xi}'_{j,s^j}(\xi) = x(\tau_j), && \bar{\Xi}'_{j,q^j}(\rho) = y(\tau_j), && \bar{\Xi}'_{j,p^{j+1}}(\xi) = w(\tau_j). \end{aligned}$$

6.2.2 The GMRES Method for the Solution of the Linearized System

As in the indirect multiple shooting approach, the pseudo matrix notation of problem (6.28) has a *sparse tridiagonal block structure*, but with more complicated blocks on the diagonal. Nevertheless, the application of the *generalized minimum residual method* is an obvious choice, and the application of a preconditioner is absolutely necessary. This fact is outlined by consideration of Example 6.2 which has been solved for differently fine discretizations and a different number of intervals as shown in Figure 6.11. As before a *forward backward block Gauss Seidel* preconditioner is due to the structure of the pseudo system matrix K an appropriate choice. This assumption is additionally supported by its outstanding performance in comparison with different preconditioners as shown in Figure 6.12. Again, we introduce a matrix decomposition into a lower triangular pseudo block matrix L, an upper triangular pseudo block matrix U, and a diagonal pseudo block matrix D:

$$\begin{pmatrix} \ddots & & -U \\ & D & \\ -L & & \ddots \end{pmatrix} :\equiv \begin{pmatrix} A_0 & C_0 & & & \\ B_0 & A_1 & C_1 & & \\ & \ddots & \ddots & \ddots & \\ & & B_{m-3} & A_{m-2} & C_{m-2} \\ & & & B_{m-2} & A_{m-1} \end{pmatrix}.$$

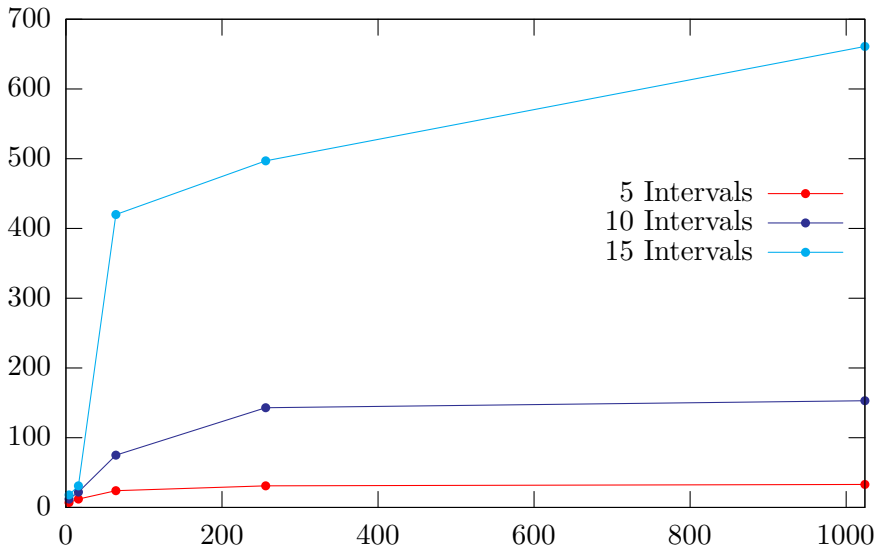


Figure 6.11: Convergence behavior of the GMRES method for differently fine discretizations calculated for different numbers of shooting intervals in Example 6.2. (x-axis: number of cells, y-axis: number of GMRES steps)

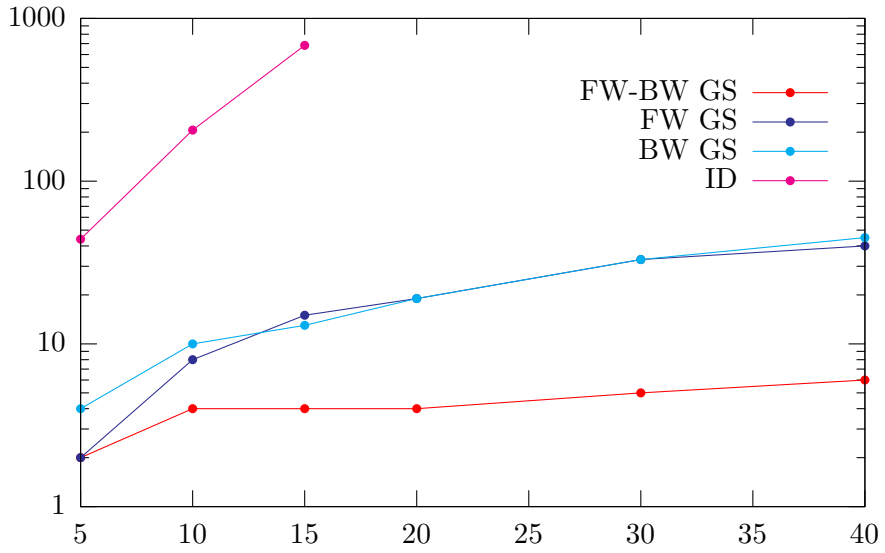


Figure 6.12: Results for the forward backward Gauss Seidel preconditioner (red), the forward Gauss Seidel preconditioner (dark blue), the backward Gauss Seidel preconditioner (blue) and without preconditioner (magenta) calculated for Example 6.2. (x-axis: number of intervals, y-axis: number of iterations)

The preconditioning pseudo matrix P for the forward backward block Gauss Seidel preconditioning was already introduced in the previous section in equation (6.5) as

$$P := (D - L)D^{-1}(D - U),$$

and the preconditioned problem is given as in equation (6.6) by the linear system

$$P^{-1}Kx = P^{-1}b.$$

While the general setup of the GMRES method is the same as in Algorithm 6.2, the different block structure of the diagonal blocks results in a more sophisticated block inversion in the case of direct multiple shooting. There are different possible ways to invert a diagonal block of the pseudo matrix D. On the one hand, a diagonal block of D can be interpreted as an optimization problem. In this case, standard techniques for the solution of optimization problems can be applied. On the other hand the inversion can be achieved by direct application of an iterative solver. The interpretation in terms of an optimization problem is thoroughly discussed in [20], where different examples and numerical results in the context of linear quadratic optimization problems are given. We follow the other way and apply an iterative solver for the inversion of the diagonal block. But first, let us describe for the sake of completeness the multiplication with the preconditioning pseudo block matrix by successive substitution. As mentioned, the procedure of preconditioning is mainly performed as before except for the additional multiplication with D. First, the inversion of the lower diagonal pseudo block matrix $D - L$ is done by forward substitution, as shown in equation (6.29). Afterwards, we multiply with the diagonal pseudo block matrix D and finally the inversion of the upper triangular pseudo block matrix $D - U$ is obtained by backward substitution as given in equation (6.30).

$$(D-L)x = r \Leftrightarrow \left\{ \begin{array}{l} x_0 = A_0^{-1} r_0 \\ x_1 = A_1^{-1}(r_1 - B_0 x_0) \\ \quad \vdots \\ x_{m-2} = A_{m-2}^{-1}(r_{m-2} - B_{m-3} x_{m-3}) \\ x_{m-1} = A_{m-1}^{-1}(r_{m-1} - B_{m-2} x_{m-2}) \end{array} \right\} \tag{6.29}$$

$$(D-U)x = r \Leftrightarrow \left\{ \begin{array}{l} x_{m-1} = A_{m-1}^{-1} r_{m-1} \\ x_{m-2} = A_{m-2}^{-1}(r_{m-2} - C_{m-2} x_{m-1}) \\ \quad \vdots \\ x_1 = A_1^{-1}(r_1 - C_1 x_2) \\ x_0 = A_0^{-1}(r_0 - C_0 x_1) \end{array} \right\} \tag{6.30}$$

The evaluation of the directional derivatives has already been discussed in the prequel. We now describe the block inversion of A_j by means of an iterative solver. We decided to apply a standard GMRES method as already stated in Algorithm 6.2 for the solution of the linear block problem, where the pseudo matrix vector product $y = A_j x$ is obtained by solution of additional problems as given in Algorithm 6.11.

Algorithm 6.11 Evaluation of the pseudo matrix vector product for a diagonal block A_j.

1: **if** $j = 0$ **then**
2: Calculate $v_0 \in H_h^1$ from $(v_0, \varphi) = (\bar{S}'_{0,q^0}(x_1), \varphi) \quad \forall \varphi \in H_h^1$.
3: Calculate $v_1 \in Q_d^1$ from $(v_1, \chi) = \alpha_3(x_1, \chi)_{Q^j} - b''_{qq}(q_\sigma^0)(x_1, \chi, z_\sigma^0) \quad \forall \chi \in Q_d^1$.
4: Set $y_0 = -x_0 + v_0$.
5: Set $y_1 = v_1$.
6: **return** y.
7: **else if** $j > 0$ **and** $j < m - 1$ **then**
8: Calculate $v_0 \in H_h^{j+1}$ from $(v_0, \varphi) = \bar{S}'_{j,q^j}(x_1) \quad \forall \varphi \in H_h^{j+1}$.
9: Calculate $v_2 \in H_h^j$ from $(v_2, \psi) = (\bar{\Xi}'_{jq^j}(x_2), \psi) \quad \forall \psi \in H_h^j$.
10: Calculate $v_1 \in Q_d^j$ from $(v_1, \chi) = \alpha_3(x_1, \chi)_{Q^j} - b''_{qq}(q_\sigma^j)(x_1, \chi, z_\sigma^j) \quad \forall \chi \in Q_d^j$.
11: Set $y_0 = -x_0 + v_0$.
12: Set $y_1 = v_1$.
13: Set $y_2 = -x_2 + v_2$.
14: **return** y.
15: **else**
16: Calculate $v_1 \in H_h^{m-1}$ from $(v_1, \psi) = (\bar{\Xi}'_{jq^{m-1}}(x_0), \psi) \quad \forall \psi \in Q_d^{m-1}$.
17: Calculate $v_0 \in Q_d^{m-1}$ from $(v_0, \chi) = \alpha_3(x_0, \chi)_{Q^j} - b''_{qq}(q^{m-1})(x_0, \chi, z^j) \quad \forall \chi \in Q_d^{m-1}$.
18: Set $y_0 = v_0$.
19: Set $y_1 = -x_1 + v_1$.
20: **return** y.

Remark 6.15. The conversion of the algorithm to a vector based one is straightforward, and can be achieved by consideration of Remark 6.14 for the interpretation of the solutions and solution operators in terms of vectors and matrices.

We want to give an impression of the additional effort which is needed for the preconditioning of the system. Therefore, we assume the best case, which is that we are able to invert the diagonal block A_j in one step of the inner GMRES method. The analysis of the costs yields the following coherence: 1 step of the plain GMRES method needs the numerical solution of $2 \cdot m$ initial value problems, whereas 1 step of the preconditioned GMRES method needs the solution of at least $6 \cdot m$ initial value problems. Thus, a compensation of the additional effort is achieved only if the number of GMRES iterations is at least reduced to 1/3 of those in the unpreconditioned case. The best case usually does not hold for our applications, such that an enormously larger number of additional initial value problems has to be solved than predicted by this simple formula. Nevertheless, the reduction obtained by preconditioning still reduces the overall effort and enables us to solve those problems for which no convergence could be obtained without preconditioning. We have solved Example 6.2 with and without preconditioning on different numbers of intervals and have plotted the number of solved initial value problems for different numbers of multiple shooting intervals in Figure 6.13.

Another promising approach is the application of condensing techniques similar to the ODE approach. We give a detailed description of this idea in the next subsection.

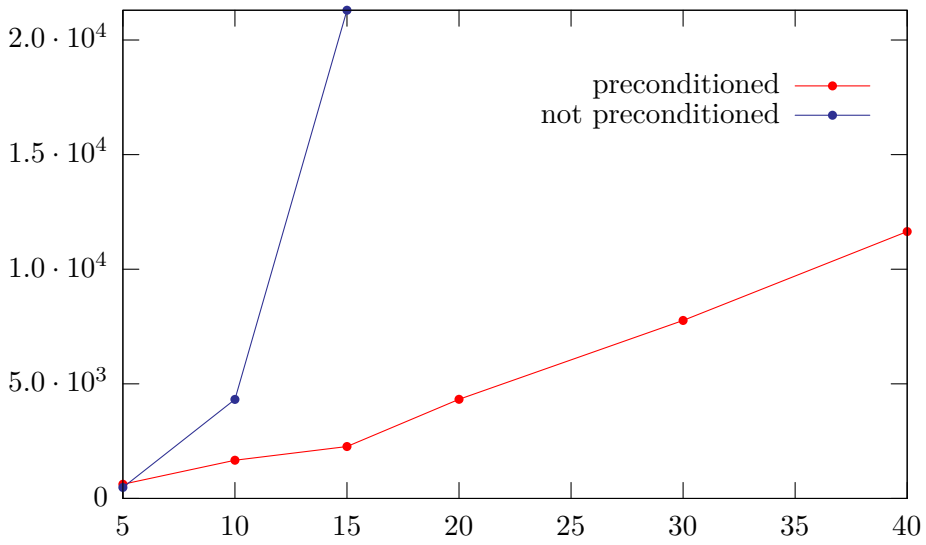


Figure 6.13: Number of solved initial value problems for different numbers of shooting intervals, calculated with and without preconditioning. (x-axis: number of intervals, y-axis: number of solved problems)

6.2.3 Condensing Techniques for the Solution of the Linearized System

We recall the formulation of the linearized system (6.26) and aim at the *reduction* of this system *onto the space of the control variables* by successive substitution. For this purpose, we make use of the following relations.

$$
\begin{aligned}
(\Delta s_h^1, \varphi) &= (\bar{S}'_{jq^0}(\Delta q_\sigma^0), \varphi) - (r_u^0, \varphi), \\
(\Delta s_h^2, \varphi) &= (\bar{S}'_{js^1}(\Delta s_h^1) + \bar{S}'_{jq^1}(\Delta q_\sigma^1), \varphi) - (r_u^1, \varphi), \\
&\vdots \\
(\Delta s_h^{m-1}, \varphi) &= (\bar{S}'_{m-2s^{m-2}}(\Delta s_h^{m-2}) + \bar{S}'_{m-2q_\sigma^{m-2}}(\Delta q_\sigma^{m-2}), \varphi) - (r_u^{m-2}, \varphi),
\end{aligned}
$$

$$
\begin{aligned}
(\Delta p_h^{m-1}, \psi) &= (\bar{\Xi}'_{js^{m-1}}(\Delta s_h^{m-1}) + \bar{\Xi}'_{jq^{m-1}}(\Delta q_\sigma^{m-1}), \psi) - (r_z^{m-1}, \psi), \\
(\Delta p_h^{m-2}, \psi) &= (\bar{\Xi}'_{jp^{m-1}}(\Delta p_h^{m-1}) + \bar{\Xi}'_{js^{m-2}}(\Delta s_h^{m-2}) - \bar{\Xi}'_{jq_\sigma^{m-2}}(\Delta q_\sigma^{m-2}), \psi) - (r_z^{m-2}, \psi), \\
&\vdots \\
(\Delta p_h^1, \psi) &= (\bar{\Xi}'_{jp^2}(\Delta p_h^2) + \bar{\Xi}'_{js^1}(\Delta s_h^1) + \bar{\Xi}'_{jq^1}(\Delta q_\sigma^1), \psi) - (r_z^1, \psi).
\end{aligned}
$$

We can now write the unknown Newton increments for primal and dual shooting variables as a function of the control variables. Introducing

$$
\begin{aligned}
f_j &: \tilde{Q}_d \to H_h^j, \\
g_j &: \tilde{Q}_d \to H_h^j
\end{aligned}
$$

such that

$$\begin{aligned} \Delta s_h^j &= f_j(\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1}) \\ \Delta p_h^j &= g_j(\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1}) \end{aligned}'$$

we reduce the linear multiple shooting system (6.26) to the determination of the control increments $\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1}$ from the condensed multiple shooting system (6.31).

Let for $j = 0, \ldots, m-1$ the following equations be fulfilled:

$$\begin{aligned} \alpha_3(\Delta q_\sigma^j, \chi)_{Q^j} - b''_{qq}(q_\sigma^j)(\Delta q_\sigma^j, \chi, \Xi_j(p_h^{j+1}, s_h^j, q_\sigma^j)) \\ - b'_q(q_\sigma^j)(\chi, \Xi'_{js^j}(f_j(\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1})) + \Xi'_{jq^j}(\Delta q_\sigma^j)) \\ + b'_q(q_\sigma^j)(\chi, \Xi'_{jp^{j+1}}(g_j(\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1}))) = -(r_q^j, \chi)_{Q^j} \quad \forall \chi \in Q_d^j. \end{aligned} \tag{6.31}$$

The whole solution procedure of each Newton step is thus brought down to the solution of (6.31) by application of a GMRES method. The algorithmic performance of this GMRES method needs the evaluation of the directional derivatives given on the left-hand side of (6.31), which is done as stated in Algorithm 6.12.

Algorithm 6.12 Calculation of the directional derivatives v_j in the GMRES method

Require: Iterates $\Delta q_\sigma^0, \ldots, q_\sigma^{m-1}$.

1: **for** $j = 1, \ldots, m-1$ **do**
2: Evaluate $\Delta s_h^j = f_j(\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1})$.
3: **for** $j = m-1, \ldots, 1$ **do**
4: Evaluate $\Delta p_h^j = g_j(\Delta q_\sigma^0, \ldots, \Delta q_\sigma^{m-1})$.
5: **for** $j = 0, \ldots m-1$ **do**
6: Calculate directional derivative $v_j \in Q_d^j$ from

$$\begin{aligned} (v_j, \chi) = \alpha_3(\Delta q_\sigma^j, \chi)_{Q^j} - b''_{qq}(q_\sigma^j)(\Delta q_\sigma^j, \chi, \Xi_j(p_h^{j+1}, s_h^j, q_h^j)) \\ - b'_q(q_\sigma^j)(\chi, \Xi'_{js^j}(\Delta s_h^j) + \Xi'_{jq^j}(\Delta q_\sigma^j) + \Xi'_{jp^{j+1}}(\Delta p_h^{j+1})) \quad \forall \chi \in Q_d^j. \end{aligned}$$

7: **return** $v_j \in \tilde{Q}_d$.

Remark 6.16. (Algorithm 6.12 in terms of the coefficient vectors) The algorithm can explicitely be written in matrix vector notation by applying the notation introduced in Remark 6.14. For example with the vectors defined as in Remark 6.14 we rewrite step 6 as

$$\zeta_j = -\mathcal{C}_2 \Delta\xi^{j+1,p} - \mathcal{B}_2 \Delta\xi^{j,s} - \mathcal{A}_2 \Delta\xi^{j,q}$$

where the coefficient vector of the directional derivative is denoted as $\zeta_j \in \mathbb{R}^{\nu_j}$.

The reduction of the number of GMRES steps by condensing the system is quite remarkable and the additional effort is limited to the condensing in the beginning and the expansion after convergence. These procedures make a total amount of $4 \cdot m$ additional initial value problems. Figure 6.14 gives an impression of the improvement obtained by application of the

condensing techniques in comparison to the preconditioned direct multiple shooting approach. Example 6.2 has been solved on 1024 cells by both techniques, and the number of solved initial value problems gives evidence of the computational effort saved by condensing.

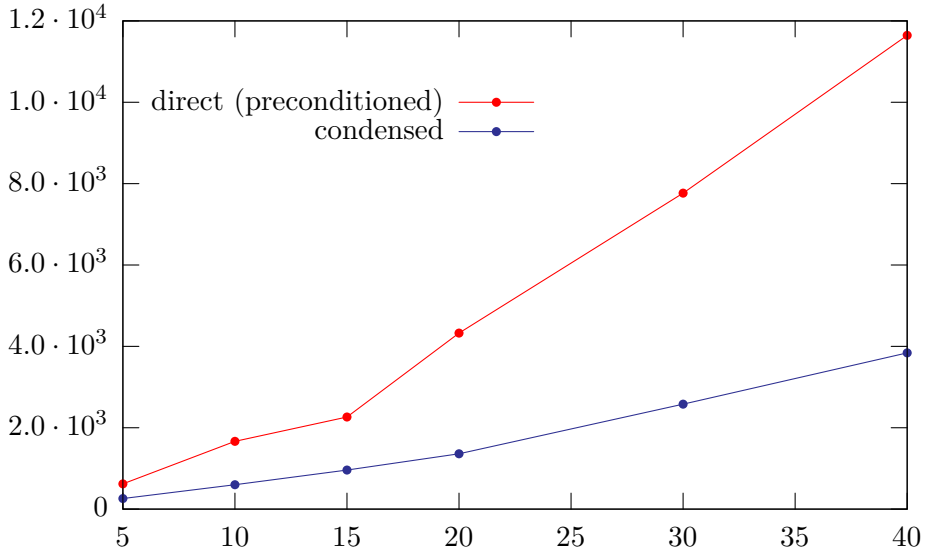


Figure 6.14: Number of solved initial value problems for different numbers of shooting intervals, calculated for preconditioned and condensed direct multiple shooting. (x-axis: number of intervals, y-axis: number of solved problems)

As mentioned in the introduction of this section, the condensed multiple shooting approach comes close to the idea of multiple shooting for ODE constrained optimization problems. At this point, we are able to underline the restrictions of the transfer of the direct multiple shooting approach from ODEs to PDEs. This is done in the following subsection.

6.2.4 From ODEs to PDEs – Limitations

In Chapter 3 we have given an overview of state-of-the-art techniques for the solution of the direct multiple shooting approach for ODE constrained optimization. The main advantage of multiple shooting for ODEs is the high efficiency which is not only obtained by application of condensing techniques, but depends crucially on the efficiency of the time stepping schemes and the simultaneous calculation of all directional derivatives for the explicit assembling and inversion of the large matrix of all directional derivatives.

In the context of PDEs not all of the latter features can be applied for certain reasons, and further research is necessary. First of all, we discretize in time by discontinuous or continuous Galerkin methods, and are thus limited to the application of those time stepping schemes which can be interpreted as a Galerkin method. Numerical computations in this thesis have only been performed for the $dG(0)$ first order time stepping scheme. Further investigations are needed to test higher order time stepping schemes and the possible improvement of the multiple shooting approach.

Second, the simultaneous calculation of all directional derivatives for the assembling of the large system matrix results in an unacceptable high computational effort whenever the control is discretized sufficiently fine. Consider for example the case of a distributed control on a spatial mesh of 5000 nodes, and the states discretized accordingly. On each interval I_j, the directional derivatives with respect to q^j, s^j and λ^j have to be calculated for every direction, that is with respect to every basis vector of the finite element discretization. Assuming bilinear elements, this makes a total of 5000 forward and 5000 backward initial value problems per interval. And for a total of 10 intervals, we obtain an amount of 100000 initial value problems for the assembling of the matrix.

Third, the storage requirements are extremely large. The system matrix for the condensed system in the above mentioned example is of the size of $10 \cdot 5000 \times 10 \cdot 5000 = 50000 \times 50000$ and (due to the condensing) usually not of sparse structure. If the matrix is once assembled, the explicit inversion is quite expensive too, and consequently, this procedure is not suitable in the context of PDEs with fine spatial discretizations of the control space.

Finally, we want to devote the rest of this chapter to a numerical comparison of the efficiency of the previously presented multiple shooting approaches for PDE constrained optimization problems.

6.3 Numerical Comparison of the Direct and Indirect Multiple Shooting Approach

In this final section, a numerical comparison between the indirect multiple shooting approach, the direct multiple shooting approach and the condensed multiple shooting approach is given. We consider a nonlinear optimization problem with Neumann boundary control.

Example 6.5. (Neumann boundary control on a T-shaped domain) We search to minimize the distributed cost functional

$$\min_{(q,u)\in Q\times X} J(q,u) = \frac{1}{2}\int_I \|u-\bar{u}\|^2_{\Gamma_o} dt + \frac{\alpha}{2}\int_I \|q\|^2_Q dt$$

subject to the constraining nonlinear equation

$$\begin{aligned} \partial_t u - \Delta u - u^3 - u = 0 \qquad &\text{on} \quad I\times\Omega_T, \\ u(0) = u_0 \qquad &\text{on} \quad \Omega_T, \\ \partial_n u = q \qquad &\text{on} \quad I\times\Gamma_c, \\ \partial_n u = 0 \qquad &\text{on} \quad I\times\Omega_T\backslash\Gamma_c. \end{aligned}$$

Ω_T is a T-shaped domain with control boundary Γ_c and observation boundary Γ_o as sketched in Figure 6.15. In the concrete case of interest, we pick the parameters $u_0 = 0.9$, $\bar{u} \equiv 1$, $\alpha = 0.01$ and the time domain $I = (0, 2)$. Furthermore, the time stepping scheme is initialized with a time step size of 0.01.

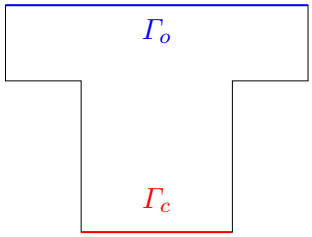


Figure 6.15: Ω_T with observation and control boundary (Γ_o and Γ_c).

The numerical comparison of the efficiency of the three approaches is mainly based on time measurements – first of all the total computation time of the approaches gives evidence of the efficency of each approach, but additionally we point out, what the time is spent for in detail. Throughout this section, we consider the solution for 5, 10, 15 and 20 multiple shooting intervals. The controls are intervalwise constant in time but distributed in space. The values of all time measurements are given in seconds.

Remark 6.17. The following computations are performed on only one fixed spatial mesh. Nevertheless, numerical computations on meshes of different refinement levels suggest the assumption that the results presented in this section are independent of the mesh size.

The total time needed for the solution of Example 6.5 by the different approaches is depicted against the number of intervals in Figure 6.16. While the time needed for the solution of the

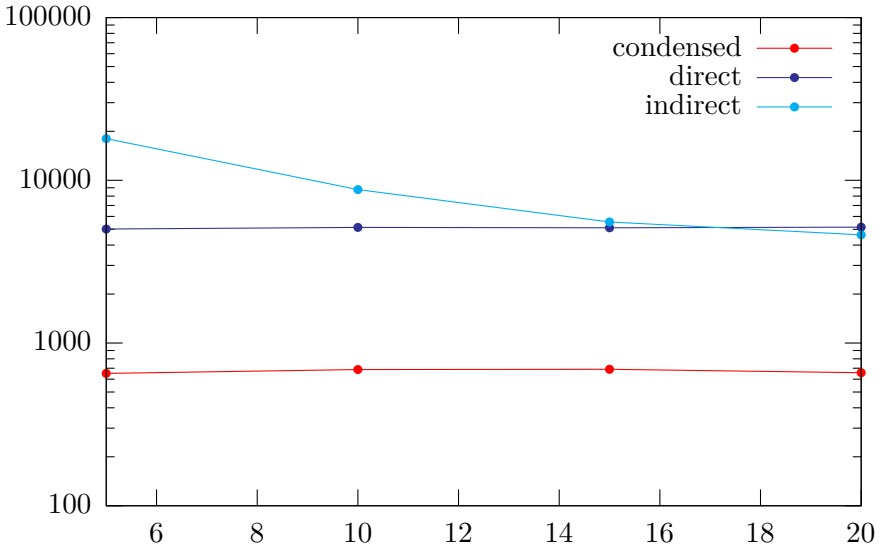


Figure 6.16: Total time for different numbers of intervals for Example 6.5, solved by different multiple shooting approaches. (x-axis: number of intervals, y-axis: time in seconds)

initial value problems is inverse proportional to the number of intervals, the total number of initial value problems grows proportional to the number of intervals. Furthermore, the

number of steps of the outer Newton method and the outer GMRES method are nearly independent from the number of intervals, such that the approximately constant total time for direct and condensed approach is the logical consequence. On the other hand, for the indirect approach, the number of iterative steps needed for the solution of the intervalwise boundary value problems decreases with the interval length, while the relations above still hold. Therefore, the total time needed for the solution of the problem by the indirect approach decreases with a growing number of intervals.

The distribution of the total time onto the solution of the linear and nonlinear interval problems for different multiple shooting parameterizations is shown in Figure 6.17. The

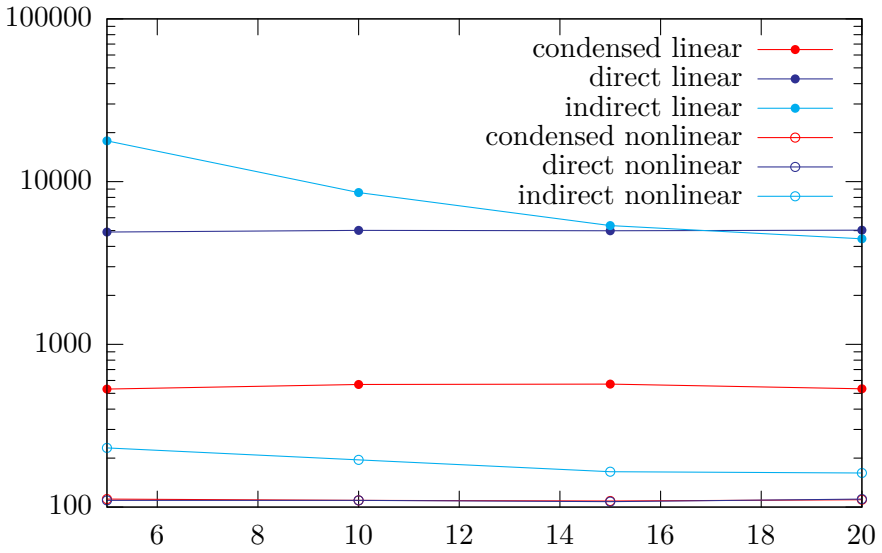


Figure 6.17: Time spent for the solution of the linear and the nonlinear interval problems for Example 6.5, solved by different multiple shooting approaches. (x-axis: number of intervals, y-axis: time in seconds)

plots confirm what we have already postulated in the previous sections: For all approaches, most of the time is spend for the solution of the linearized system of matching conditions. Precisely, Figures 6.18 and 6.19 point out that for direct and indirect multiple shooting without condensing about 97% of the computation time are needed for this procedure, while only approximately 2% pass during the calculation of the residual. The rest of the time is spent for the initialization, the output, and other procedures which do not directly belong to the multiple shooting method. For the condensed approach the time distribution is more equilibrated, though still about 82% of the time is needed for the solution of the linearized problem while only 17% of the time are spent for the calculation of the residual.

So far, we have seen, that the condensed approach is the most promising one out of the three different approaches considered in this chapter. In the next two chapters, we want to combine the multiple shooting approach with mesh adaptation gained by a modification of the dual weighted residual method. This idea allows us to the increase the efficency of the methods by

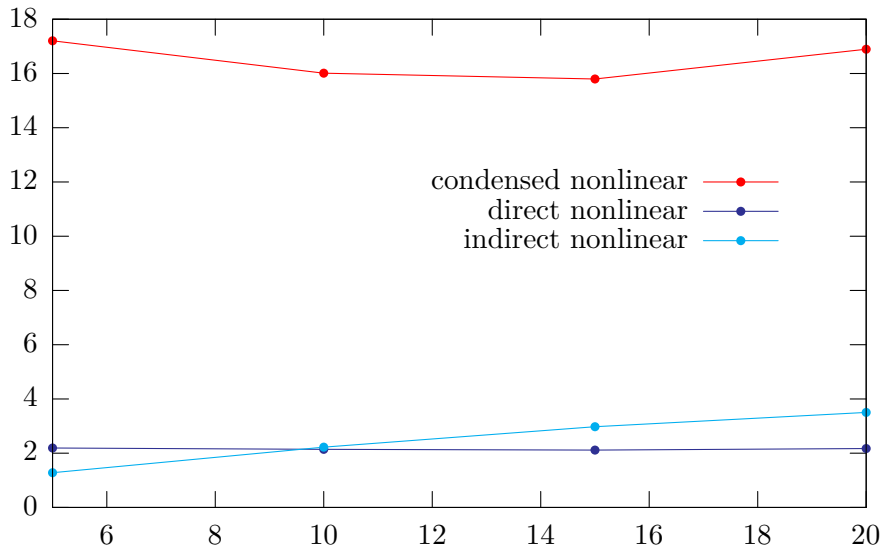


Figure 6.18: Percentage of the time for the solution of the nonlinear interval problems for Example 6.5, solved by different multiple shooting approaches. (x-axis: number of intervals, y-axis: percentage of time)

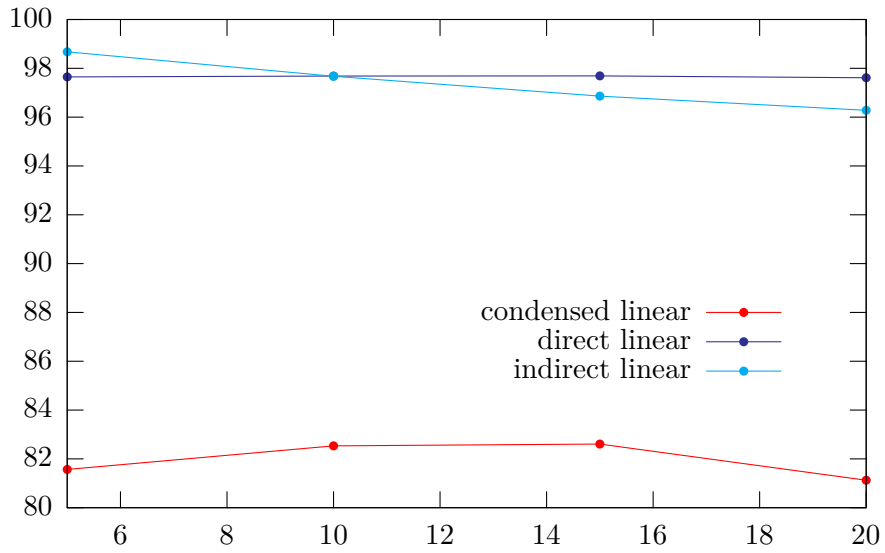


Figure 6.19: Percentage of the time for the solution of the linear interval problems for Example 6.5, solved by different multiple shooting approaches. (x-axis: number of intervals, y-axis: percentage of time)

solving the PDEs under consideration on meshes that are as coarse as possible for obtaining a certain discretization error in the cost functional.

7 A Posteriori Error Estimation

This chapter is devoted to the presentation of a posteriori error estimation. Starting with the repetition of the original ideas of *dual weighted residual a posteriori error estimation* (DWR method) in Section 7.1, we proceed with the application of the classical error estimator to the multiple shooting approach. We point out, that the classical error estimator is limited to the case of coinciding adjacent meshes at the multiple shooting nodes. Next, a new error estimator, especially suited for the application to the multiple shooting solution, is derived in Section 7.2. This error estimator is designed to incorporate the *projection errors* on the multiple shooting nodes due to different adjacent meshes. It is based on the consideration of the converged multiple shooting system in terms of a Galerkin approach. While the error estimators are developed for the error between the fully discrete solution and the continuous solution in function space, for the numerical consideration we limit ourselves to the further investigation of the spatial part of the error for the remaining part of the chapter. The developed error estimators contain the unknown solution of the optimization problem. Therfore, the evaluation of the error estimators needs reliable approximations of the aforementioned solutions. Consequently, the practical evaluation of the error is discussed in Section 7.3. Finally, this chapter closes by presenting some numerical examples in Section 7.4 which illustrate the performance of the developed error estimators.

Remark 7.1. We should remark that the application of the considered error estimators to the multiple shooting solution is independent of the multiple shooting approach. The error estimator as an additional feature is applied to the solution after convergence and needs no information on the approach itself.

7.1 The Classical Error Estimator for the Cost Functional

We start this section by reviewing a modification of a fundamental result from [7]. This modification is for example derived in [29] and is an essential ingredient for all later developments.

Lemma 7.1. *Let Y be a function space and L a three times Gâteaux differentiable functional on Y. We seek a stationary point y^* of L on $Y_1 \subset Y$, that is*

$$L'(y_1^*)(y_1) = 0 \quad \forall y_1 \in Y_1.$$

This equation is approximated by a Galerkin method, using a space $Y_2 \subset Y$. The approximative problem seeks $y_2^* \in Y_2$ such that

$$L'(y_2^*)(y_2) = 0 \quad \forall y_2 \in Y_2. \tag{7.1}$$

If the continuous solution y_1^* fulfills additionally

$$L'(y_1^*)(y_2) = 0 \quad \forall y_2 \in Y_2, \tag{7.2}$$

and the following error representation holds for arbitrary $y_2 \in Y_2$.

$$L(y_1^*) - L(y_2^*) = \frac{1}{2} L'(y_2^*)(y_1^* - y_2) + \mathcal{R}.$$

The remainder term $\mathcal{R}$ is given by

$$\mathcal{R} = \frac{1}{2} \int_0^1 L'''(y_2^* + se)(e,e,e) \cdot s \cdot (s-1) ds$$

where the error e is defined as $e = y_1^* - y_2^*$.

Proof. First, from the main theorem of calculus, the following identity holds for $y_1 \in Y_1$, $y_2 \in Y_2$:

$$L(y_1^*) - L(y_2^*) = \int_0^1 L'(y_2^* + se)(e) ds.$$

Application of the trapezoidal rule to the integral yields:

$$\int_0^1 L'(y_2^* + se)(e) ds = \frac{1}{2} L'(y_2^*)(e) + \frac{1}{2} L'(y_1^*)(e) + \underbrace{\frac{1}{2} \int_0^1 L'''(y_2^* + se)(e,e,e) \cdot s \cdot (s-1) ds}_{=:\mathcal{R}}.$$

The second addend vanishes according to (7.2), and due to (7.1) we have for arbitrary $y_2 \in Y_2$

$$\frac{1}{2} L'(y_2^*)(e) = \frac{1}{2} L'(y_2^*)(y_1^* - y_2).$$

This completes the proof. □

Remark 7.2. (Notation) Throughout this chapter different Lagrangians, cost functionals, and residuals are defined. In order to keep the notation easily understandable we use the same notation within the different subsections whereas the certain meaning becomes clear in the context. The Lagrangians in function space are denoted by $\mathcal{L}$, Lagrangians of the discretized problems by $\tilde{\mathcal{L}}$. Furthermore, all residuals are named $\tilde{\rho}$ with, if necessary, additional indices to specify the corresponding interval number and equation.

We want to recall some results on the estimation of the discretization error in the cost functional by the dual weighted residual method. The idea of this approach was first developed by Becker, Kapp, and Rannacher in [4], [5], [3], [7] and extensions to optimization problems have been developed by Meidner and Vexler in [6], [30]. Whereas Meidner and Vexler are interested in splitting the discretization error of the fully discretized solution into different parts due to the spatial discretization, the temporal discretization, and the discretization of the control space, we first derive the error estimators for the error $J(q,u) - J(q_\sigma, u_\sigma)$ and later in the numerical applications restrict ourselves to the case of estimating $J(q_k, u_k) - J(q_\sigma, u_\sigma)$.

We start by recalling a result for the DWR error representation of the cost functional J on the whole time interval. Therefore, we need some preparatory results presented in the

following. We recall the definition of the Lagrangian $\mathcal{L} : Q \times X \times X \to \mathbb{R}$ from equation (4.3), which was given by

$$\mathcal{L}(q,u,z) := J(q,u) - \{((\partial_t u, z)) + a(u)(z) + b(q)(z) + (u(0), z(0)) - ((f,z)) - (u_0, z(0))\}.$$

Its discrete counterpart $\tilde{\mathcal{L}} : Q_d \times \tilde{X}_k^{r,s} \times \tilde{X}_k^{r,s} \to \mathbb{R}$ reads in terms of the chosen dG(r)cG(s) discretization

$$\tilde{\mathcal{L}}(q_\sigma, u_\sigma, z_\sigma) := J(q_\sigma, u_\sigma) - \left\{ \sum_{l=1}^{n} ((\partial_t u_\sigma, z_\sigma))_l + a(u_\sigma)(z_\sigma) + b(q_\sigma)(z_\sigma) + \sum_{l=0}^{n-1} ([u_\sigma]_l, z_{k,l}^+) + (u_{\sigma,0}^- - u_0, z_{\sigma,0}^-) - ((f, z_\sigma)) \right\}. \quad (7.3)$$

For the derivatives of the discrete Lagrangian the following relations hold true:

Lemma 7.2. *Let for the optimization problem of interest, lastly stated in* (4.2)*, the solutions of the continuous, resp. discretized, problem be given as*

$$(q,u,z) \in Q \times X \times X \quad \textit{and} \quad (q_\sigma, u_\sigma, z_\sigma) \in Q_d \times \tilde{X}_{kh}^{r,s} \times \tilde{X}_{kh}^{r,s}.$$

Then the following equations hold true:

$$\tilde{\mathcal{L}}'(q_\sigma, u_\sigma, z_\sigma)(\hat{q}_\sigma, \hat{u}_\sigma, \hat{z}_\sigma) = 0 \quad \forall (\hat{q}_\sigma, \hat{u}_\sigma, \hat{z}_\sigma) \in Q_d \times \tilde{X}_{kh}^{r,s} \times \tilde{X}_{kh}^{r,s}, \quad (7.4)$$

$$\tilde{\mathcal{L}}'(q,u,z)(\hat{q}_\sigma, \hat{u}_\sigma, \hat{z}_\sigma) = 0 \quad \forall (\hat{q}_\sigma, \hat{u}_\sigma, \hat{z}_\sigma) \in Q_d \times \tilde{X}_{kh}^{r,s} \times \tilde{X}_{kh}^{r,s}. \quad (7.5)$$

Proof. Equation (7.4) is equivalent to the first order optimality condition of the discretized optimization problem and thus fulfilled for its solution $(q_\sigma, u_\sigma, z_\sigma) \in Q_d \times \tilde{X}_{kh}^{r,s} \times \tilde{X}_{kh}^{r,s}$. The proof of equation (7.5) is more sophisticated. In order to clarify the basic ideas of this proof, we show how the verification is performed for the primal equation

$$\tilde{\mathcal{L}}_z^{j\prime}(q,u,z)(\hat{z}_\sigma) = 0 \quad \forall \hat{z}_\sigma \in \tilde{X}_{kh}^{r,s}. \quad (7.6)$$

The proof for the remaining equations

$$\tilde{\mathcal{L}}_u^{j\prime}(q,u,z)(\hat{u}_\sigma) = 0 \qquad \forall \hat{u}_\sigma \in \tilde{X}_{kh}^{r,s},$$
$$\tilde{\mathcal{L}}_q^{j\prime}(q,u,z)(\hat{q}_\sigma) = 0 \qquad \forall \hat{q}_\sigma \in Q_d.$$

follows analogously.

Regarding the definition of the Lagrangian in (7.3), the jumps and the initial condition cancel out for the continuous solution. We remain with the reformulation of (7.6) as follows:

$$\sum_{l=1}^{n_j} ((\partial_t u, \hat{z}_\sigma))_l + a(u)(\hat{z}_\sigma) + b(q)(\hat{z}_\sigma) = 0 \quad \forall \hat{z}_\sigma \in \tilde{X}_{kh}^{r,s}. \quad (7.7)$$

Now, by density of X in $L^2(I,V)$ with respect to the norm on $L^2(I,V)$, and by the inclusion $\tilde{X}_{kh}^{r,s} \subset L^2(I,V)$, we are able to approximate $\hat{z}_\sigma$ by a sequence of functions in X. Considering the limit, we obtain that (7.7) is true and consequently that equation (7.6) holds true. □

Now, we have everything at hand to consider the error representation of the functional error in the cost functional J:

$$J(q,u) - J(q_\sigma, u_\sigma).$$

We are able to prove the following well known result for the discretization error:

Theorem 7.3. *Let (q,u,z) be a stationary point of the Lagrangian $\mathcal{L}$, that is, it fulfills the first order optimality condition (4.4). Furthermore, let $(q_\sigma, u_\sigma, z_\sigma)$ be a stationary point of the discrete Lagrangian $\tilde{\mathcal{L}}$, that is, it solves the discretization of (4.4). For the error of the cost functional the following representation holds:*

$$J(q,u) - J(q_\sigma, u_\sigma) = \frac{1}{2}\tilde{\mathcal{L}}'(q_\sigma, u_\sigma, z_\sigma)(q - \hat{q}_\sigma, u - \hat{u}_\sigma, z - \hat{z}_\sigma) + R_\sigma \tag{7.8}$$

where $(\hat{q}_\sigma, \hat{u}_\sigma, \hat{z}_\sigma) \in Q_d \times X_{kh}^{r,s} \times X_{kh}^{r,s}$ arbitrary and the remainder terms R_σ has the same form as in Lemma 7.1 with $L = \tilde{\mathcal{L}}^j$.

Proof. From previous thoughts we know, that the continuous solution fulfills (4.4), and the fully discrete solution fulfills its discretization. Therefore, the primal equation in the formulation of the Lagrangian vanishes and we directly obtain

$$J(q,u) - J(q_\sigma, u_\sigma) = \tilde{\mathcal{L}}(q,u,z) - \tilde{\mathcal{L}}(q_\sigma, u_\sigma, z_\sigma).$$

The remaining part of the proof is obtained from Lemma 7.1 with

$$\begin{aligned} Y_1 &:= Q \times X \times X, \\ Y_2 &:= Q_d \times \tilde{X}_{kh}^{r,s} \times \tilde{X}_{kh}^{r,s}, \\ Y &:= Y_1 + Y_2. \end{aligned}$$

Due to Lemma 7.2, condition (7.2) in Lemma 7.1 holds for $L = \tilde{\mathcal{L}}$, and we retrieve the equality

$$\tilde{\mathcal{L}}(q,u,z) - \tilde{\mathcal{L}}(q_\sigma, u_\sigma, z_\sigma) = \frac{1}{2}\tilde{\mathcal{L}}'(q_\sigma, u_\sigma, z_\sigma)(q - \hat{q}_\sigma, u - \hat{u}_\sigma, z - \hat{z}_\sigma) + R_\sigma$$

for arbitrary $(\hat{q}_\sigma, \hat{u}_\sigma, \hat{z}_\sigma) \in Q_d \times X_{kh}^{r,s} \times X_{kh}^{r,s}$. Altogether this yields the stated condition

$$J(q,u) - J(q_\sigma, u_\sigma) = \frac{1}{2}\tilde{\mathcal{L}}'(q_\sigma, u_\sigma, z_\sigma)(q - \hat{q}_\sigma, u - \hat{u}_\sigma, z - \hat{z}_\sigma) + R_\sigma$$

and completes the proof. □

We can furthermore write the error identities of Theorem 7.3 by means of the residuals $\tilde{\rho}_u(q,u)(\varphi)$, $\tilde{\rho}_z(q,u,z)(\psi)$, $\tilde{\rho}_q(q,u,z)(\chi)$ of the discretized primal, dual, and control equation:

$$\begin{aligned} \tilde{\rho}_u(q_\sigma, u_\sigma)(\varphi) &= \tilde{\mathcal{L}}_z'(q_\sigma, u_\sigma, z_\sigma)(\varphi), \\ \tilde{\rho}_z(q_\sigma, u_\sigma, z_\sigma)(\psi) &= \tilde{\mathcal{L}}_u'(q_\sigma, u_\sigma, z_\sigma)(\psi), \\ \tilde{\rho}_q(q_\sigma, u_\sigma, z_\sigma)(\chi) &= \tilde{\mathcal{L}}_q'(q_\sigma, u_\sigma, z_\sigma)(\chi). \end{aligned}$$

Introducing this notation and neglecting the remainder terms, system (7.8) turns into

$$\begin{aligned} &J(q,u) - J(q_\sigma, u_\sigma) \\ &\qquad \approx \frac{1}{2}\left\{\tilde{\rho}_u(q_\sigma, u_\sigma)(z - \hat{z}_\sigma) + \tilde{\rho}_z(q_\sigma, u_\sigma, z_\sigma)(u - \hat{u}_\sigma) + \tilde{\rho}_q(q_\sigma, u_\sigma, z_\sigma)(q - \hat{q}_\sigma)\right\}. \end{aligned} \tag{7.9}$$

Example 7.1. (Residuals for the cG(s)dG(0) discretization) To make these theoretical results more substantial, the residuals for the cG(s)dG(0) discretization are stated in the following. The notation is according to equation (5.9).

$$\begin{aligned}
\tilde{\rho}_u(q_\sigma, u_\sigma)(\psi) &= (U_{\sigma,0} - u_0, \Psi_0) \\
&\quad + \sum_{l=1}^{n} \{(U_{\sigma,l} - U_{\sigma,l-1}, \Psi_l) \\
&\qquad + k_l \bar{a}(U_{\sigma,l})(\Psi_l) + k_l \bar{b}(Q_{\sigma,l})(\Psi_l) - k_l(f(t_l), \Psi_l)\Big\} \\
\tilde{\rho}_z(q_\sigma, u_\sigma, z_\sigma)(\psi) &= (Z_{\sigma,0} - Z_{\sigma,1}, \Psi_0) \\
&\quad + \sum_{l=1}^{n-1} \{(Z_{\sigma,l} - Z_{\sigma,l+1}, \Psi_l) + k_l \bar{a}'_u(U_{\sigma,l})(\Psi_l, Z_{\sigma,l}) \\
&\qquad - k_l F'(U_{\sigma,l})(\Psi_l)\} + (Z_{\sigma,n}, \Psi_n) - J'_u(Q_{\sigma,n}, U_{\sigma,n})(\Psi_n) \\
\tilde{\rho}_q(q_\sigma, u_\sigma, z_\sigma)(\psi) &= \sum_{l=1}^{n} \Big\{\bar{b}'_q(Q_{\sigma,l})(\Psi_l, Z_{\sigma,l}) - (Q_{\sigma,l}, \Psi_l)_R\Big\}
\end{aligned}$$

We continue by embedding the error estimator into the context of multiple shooting. Therefore, let us first state two preliminary remarks.

Remark 7.3. (Discretization of the multiple shooting variables) In the following, we assume that adjacent meshes at the multiple shooting nodes are equivalent for reasons of consistency. The multiple shooting variables are discretized accordingly, and for the spatial discretization we have $V_h^{j,n_j,s} = V_h^{j+1,0,s}$.

Remark 7.4. The multiple shooting solution on the different stages of discretization fulfills the matching conditions. Therefore, according to Remark 7.3, we have the following identities for the fully discrete multiple shooting solution:

$$u_{\sigma,n_j}^{j-} = u_{\sigma,0}^{(j+1)-} = s_h^{j+1} \quad \text{and} \quad z_{\sigma,0}^{(j+1)-} = z_{\sigma,n_j}^{j-} = \lambda_h^{j+1}.$$

We prove the following theorem which displays the classical DWR error representation of the functional error in terms of the intervalwise residuals:

Theorem 7.4. *Let (q,u,z), $(q_\sigma, u_\sigma, z_\sigma)$ be solutions of problem* (4.2) *on the different stages of discretization, and let us denote their intervalwise restrictions as before. Let $\tilde{\rho}_u$, $\tilde{\rho}_z$, $\tilde{\rho}_q$ denote the discrete primal, dual, and control residual of the equations in* (5.7)*. For the functional error there holds*

$$\begin{aligned}
&J(q,u) - J(q_\sigma, u_\sigma) \\
&\qquad = \frac{1}{2}\{\tilde{\rho}_u(q_\sigma, u_\sigma)(z - \hat{z}_\sigma) + \tilde{\rho}_z(q_\sigma, u_\sigma, z_\sigma)(u - \hat{u}_\sigma) + \tilde{\rho}_q(q_\sigma, u_\sigma, z_\sigma)(q - \hat{q}_\sigma)\} \quad (7.11a)
\end{aligned}$$

where the residuals are obtained by summation of the intervalwise residuals:

$$\begin{aligned}
\tilde{\rho}_u(q_\sigma, u_\sigma)(z - \hat{z}_\sigma) &= \sum_{j=0}^{m-1} \tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(z^j - \hat{z}_\sigma^j), \\
\tilde{\rho}_z(q_\sigma, u_\sigma, z_\sigma)(u - \hat{u}_\sigma) &= \sum_{j=0}^{m-1} \tilde{\rho}_z^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(u^j - \hat{u}_\sigma^j), \\
\tilde{\rho}_q(q_\sigma, u_\sigma, z_\sigma)(q - \hat{q}_\sigma) &= \sum_{j=0}^{m-1} \tilde{\rho}_q^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(q^j - \hat{q}_\sigma^j).
\end{aligned} \tag{7.11b}$$

Here, the terms $\tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(\varphi)$, $\tilde{\rho}_z^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(\psi)$, and $\tilde{\rho}_q^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(\chi)$ denote the residuals of the discretized intervalwise optimality system (5.6)*.*

Proof. The first statement has already been proven in Lemma 7.3. Therefore, only the proof of the last statement remains undone. We prove the identity only for the primal residual, the result for dual and control residual follows analogously. From equation (5.2a) we retrieve by summation over the multiple shooting intervals

$$\begin{aligned}
&\sum_{j=0}^{m-1} \tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(z^j - \hat{z}_\sigma^j) = \\
&\qquad \sum_{j=0}^{m-1} \Big\{ \sum_{l=1}^{n_j} (\!(\partial_t u_\sigma^j, z^j - \hat{z}_\sigma^j)\!)_{j,l} + a(u_\sigma^j)(z^j - \hat{z}_\sigma^j) + b(q_\sigma^j)(z^j - \hat{z}_\sigma^j) \\
&\qquad\qquad + \sum_{l=0}^{n_j - 1} ([u_\sigma^j]_l, z_l^{j+} - \hat{z}_{\sigma,l}^{j+}) + (u_{\sigma,0}^{j-} - s_h^j, z_0^{j-} - \hat{z}_{\sigma,0}^{j-}) - (\!(f, z^j - \hat{z}_\sigma^j)\!)_j \Big\}.
\end{aligned}$$

Due to Remark 7.4 the terms $(u_{\sigma,0}^{j-}, z_0^{j-} - \hat{z}_{\sigma,0}^{j-})$ and $(s_h^j, z_0^{j-} - \hat{z}_{\sigma,0}^{j-})$ cancel out for $j = 1, \ldots, m-1$, and only the term for $j = 0$ remains. Furthermore, we can substitute for continuous z^j the terms $z_l^{j+} = z_l^{j-} = z(t_{j,l})$ and obtain

$$\begin{aligned}
&\sum_{j=0}^{m-1} \tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(z^j - \hat{z}_\sigma^j) = \\
&\qquad \sum_{j=0}^{m-1} \Big\{ \sum_{l=1}^{n_j} (\!(\partial_t u_\sigma^j, z^j - \hat{z}_\sigma^j)\!)_{j,l} + a(u_\sigma^j)(z^j - \hat{z}_\sigma^j) + b(q_\sigma^j)(z^j - \hat{z}_\sigma^j) \\
&\qquad + \sum_{l=0}^{n_j - 1} ([u_\sigma^j]_l, z(t_{j,l}) - \hat{z}_{\sigma,l}^{j+}) \Big\} + (u_{\sigma,0}^{0-} - u_0, z(\tau_0) - \hat{z}_{\sigma,0}^{0-}) - \sum_{j=0}^{m-1} (\!(f, z^j - \hat{z}_\sigma^j)\!)_j.
\end{aligned}$$

From Remark 7.4 we see further that for $j = 0, \ldots, m-2$ the jump terms for $l = 0$ can be written as

$$([u_\sigma^{j+1}]_0, z(\tau_{j+1}) - \hat{z}_{\sigma,0}^{(j+1)+}) = (u_{\sigma,0}^{(j+1)+} - u_{\sigma,n_j}^{j-}, z(\tau_{j+1}) - \hat{z}_{\sigma,0}^{(j+1)+}).$$

We insert this into the equation and obtain

$$\sum_{j=0}^{m-1} \tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(z^j - \hat{z}_\sigma^j) =$$
$$\sum_{j=0}^{m-1} \Big\{ \sum_{l=1}^{n_j} ((\partial_t u_\sigma^j, z^j - \hat{z}_\sigma^j))_{j,l} + a(u_\sigma^j)(z^j - \hat{z}_\sigma^j) + b(q_\sigma^j)(z^j - \hat{z}_\sigma^j)$$
$$+ \sum_{l=1}^{n_j-1} ([u_\sigma^j]_l, z(t_{j,l}) - \hat{z}_{\sigma,l}^{j+}) \Big\} + ([u_\sigma^0]_0, z(\tau_0) - \hat{z}_{\sigma,0}^{0+}) + \sum_{j=0}^{m-2} (u_{\sigma,0}^{(j+1)+} - u_{\sigma,n_j}^{j-}, z(\tau_{j+1}) - \hat{z}_{\sigma,0}^{(j+1)+})$$
$$+ (u_{\sigma,0}^{0-} - u_0, z(\tau_0) - \hat{z}_{\sigma,0}^{0-}) - \sum_{j=0}^{m-1} ((f, z^j - \hat{z}_\sigma^j))_j.$$

The right-hand side of this equation is equivalent to the residual of the discrete primal equation of (5.7):

$$\sum_{j=0}^{m-1} \tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(z^j - \hat{z}_\sigma^j) = \tilde{\rho}_u(q_\sigma, u_\sigma, z_\sigma)(z - \hat{z}_\sigma). \tag{7.12}$$

This completes the proof. □

So far we have seen how the classical DWR error estimator can be composed from intervalwise residuals under the assumption of identical adjacent meshes on the multiple shooting nodes. This assumption is due to the fact, that the solution obtained by multiple shooting must coincide with the solution from the classical approach. In the next subsection, we develop an approach that allows us to handle the error estimator for solutions with differently refined adjacent meshes. This error estimator incorporates additional projection errors on the multiple shooting nodes.

7.2 A Posteriori Error Estimation for the Multiple Shooting System

The following approach to a posteriori error estimation for the functional error is based on *embedding the whole multiple shooting approach after convergence into the context of Galerkin methods*. This procedure enables us to treat each interval separately without any assumptions on the adjacent grids from different intervals. The resulting error estimator inherits *parts of the classical DWR error estimator* but additionally incorporates *projection errors* for both primal and dual shooting variables on the multiple shooting nodes.

Remark 7.5. (Error estimation for direct and indirect multiple shooting) We derive this estimator by means of the indirect multiple shooting approach only, but remark that the obtained solutions for direct and indirect multiple shooting are identical. Therefore, the results hold accordingly for the solution of the direct multiple shooting approach.

Let us consider the following optimization problem for which we show that its optimality system is equivalent to the indirect multiple shooting formulation (4.8) with matching conditions (4.9). Using the notation as introduced in (4.10), we set up the minimization problem

$$\min_{\tilde{s},\tilde{u},\tilde{q}} \mathcal{J}(\tilde{q},\tilde{u},\tilde{s}) := \sum_{j=0}^{m-1} \left\{ \alpha_1 J_1^j(u^j) + \frac{\alpha_3}{2} \|q^j\|_{Q^j}^2 \right\} + \alpha_2 J_2(s_m) \tag{7.13a}$$

such that

1. for all $\varphi \in X^j$, $j = 0, \ldots, m-1$,

$$(\!(\partial_t u^j, \varphi)\!)_j + a(u^j)(\varphi) + b(q^j)(\varphi) + (u^j(\tau_j) - s^j, \varphi(\tau_j)) - (\!(f, \varphi)\!)_j = 0, \tag{7.13b}$$

2. for all $v \in H$, $j = 0, \ldots, m-1$

$$\begin{aligned} (s^0 - u_0, v) &= 0, \\ (s^{j+1} - u^j(\tau_{j+1}), v) &= 0. \end{aligned} \tag{7.13c}$$

The Lagrangian of (7.13), mapping from $\tilde{Q} \times \tilde{X} \times \tilde{H} \times \tilde{X} \times \tilde{H} \to \mathbb{R}$ reads

$$\begin{aligned} \mathcal{L}(\tilde{q},\tilde{u},\tilde{s},\tilde{z},\tilde{\lambda}) = &\sum_{j=0}^{m-1} \{\alpha_1 J_1^j(u^j) + \frac{\alpha_3}{2} \|q^j\|_{Q^j}^2\} + \alpha_2 J_2(s^m) \\ &- \Big\{ \sum_{j=0}^{m-1} \{(\!(\partial_t u^j, z^j)\!)_j + a(u^j)(z^j) + b(q^j)(z^j) + (u^j(\tau_j) - s^j, z^j(\tau_j)) - (\!(f, z^j)\!)_j\} \\ &\qquad + \sum_{j=0}^{m-1} \{(s^{j+1} - u^j(\tau_{j+1}), \lambda^{j+1})\} + (s^0 - u_0, \lambda^0) \Big\}. \end{aligned} \tag{7.14}$$

Explicit calculation of the first order optimality condition

$$\mathcal{L}'(\tilde{q},\tilde{u},\tilde{s},\tilde{z},\tilde{\lambda})(\tilde{\chi},\tilde{\varphi},\tilde{\xi},\tilde{\psi},\tilde{\eta}) = 0 \quad \forall (\tilde{\chi},\tilde{\varphi},\tilde{\xi},\tilde{\psi},\tilde{\eta}) \in \tilde{Q} \times \tilde{X} \times \tilde{H} \times \tilde{X} \times \tilde{H}$$

yields back the intervalwise optimality systems (4.8) of the indirect approach together with the matching conditions (4.9):

Differentiation of (7.14) with respect to the primal variables $(\tilde{u}, \tilde{s})$ yields for all $(\tilde{\varphi}, \tilde{\xi}) \in \tilde{X} \times \tilde{H}$ the condition

$$\begin{aligned} &\sum_{j=0}^{m-1} \alpha_1 J_1^{j\prime}(u^j)(\varphi^j) + \alpha_2 J_2'(s^m)(\xi^m) \\ &\qquad - \Big\{ \sum_{j=0}^{m-1} \{(\!(\partial_t \varphi^j, z^j)\!)_j + a_u'(u^j)(\varphi^j, z^j) + (\varphi^j(\tau_j) - \xi^j, z^j(\tau_j))\} \\ &\qquad\qquad + \sum_{j=0}^{m-1} \{(\xi^{j+1} - \varphi^j(\tau_{j+1}), \lambda^{j+1})\} + (\xi^0, \lambda^0) \Big\} = 0 \end{aligned}$$

which by partial integration transforms into

$$\sum_{j=0}^{m-1} \alpha_1 J_1^{j\prime}(u^j)(\varphi^j) + \alpha_2 J_2'(s^m)(\xi^m)$$
$$- \Big\{ \sum_{j=0}^{m-1} \{ -((\partial_t z^j, \varphi^j))_j + a_u'(u^j)(\varphi^j, z^j) + (z^j(\tau_{j+1}), \varphi^j(\tau_{j+1})) + (z^j(\tau_j), \xi^j)\}$$
$$+ \sum_{j=0}^{m-1} \{(\xi^{j+1} - \varphi^j(\tau_{j+1}), \lambda^{j+1})\} + (\xi^0, \lambda^0)\Big\} = 0.$$

Now, summing up the terms appropriately, we end up with the equation

$$\sum_{j=0}^{m-1} \alpha_1 J_1^{j\prime}(u^j)(\varphi^j) + \alpha_2 J_2'(s^m)(\xi^m) - (\lambda^m, \xi^m)$$
$$- \Big\{ \sum_{j=0}^{m-1} \{ -((\partial_t z^j, \varphi^j))_j + a_u'(u^j)(\varphi^j, z^j) + (z^j(\tau_{j+1}) - \lambda^{j+1}, \varphi^j(\tau_{j+1}))\}$$
$$+ \sum_{j=0}^{m-1} (\lambda^j - z^j(\tau_j), \xi^j)\Big\} = 0.$$

We separate the equations and obtain for $j = 0, \ldots, m-1$ the following PDE:

Determine $z^j \in X^j$ such that for all $\varphi^j \in X^j$

$$\alpha_1 J_1^{j\prime}(u^j)(\varphi^j) - \{ -((\partial_t z^j, \varphi^j))_j + a_u'(u^j)(\varphi^j, z^j) + (\lambda^{j+1} - z^j(\tau_{j+1}), \varphi^j(\tau_{j+1}))\} = 0.$$

This equation is equivalent to the dual equation (4.8b). Furthermore we obtain a set of conditions which are equivalent to the matching conditions for the dual variable in (4.9):

$$(z^j(\tau_j) - \lambda^j, \xi^j) = 0 \quad \forall \xi^j \in H,$$
$$\alpha_2 J_2'(s^m)(\xi^m) - (\lambda^m, \xi^m) = 0 \quad \forall \xi^m \in H.$$

Analogously we obtain by differentiation of (7.14) with respect to $\tilde{q}$ the intervalwise control equations (4.8c).

Now, as before, we are in need of the Lagrangian of the discretized problem in order to develop the error estimator for the cost functional. We denote the Lagrangian for the fully discrete problem by $\tilde{\mathcal{L}} : \tilde{Q}_d \times \tilde{X}_{kh} \times \tilde{H}_h \times \tilde{X}_{kh} \times \tilde{H}_h \to \mathbb{R}$. The discrete Lagrangian is defined by the following identity:

$$\tilde{\mathcal{L}}(\tilde{q}_\sigma, \tilde{u}_\sigma, \tilde{s}_h, \tilde{z}_\sigma, \tilde{\lambda}_h) := \sum_{j=0}^{m-1} \{\alpha_1 J_1^j(u_\sigma^j) + \frac{\alpha_3}{2}\|q_\sigma^j\|_{Q^j}^2\} + \alpha_2 J_2(s_h^m)$$
$$-\Big\{ \sum_{j=0}^{m-1} \{ \sum_{l=1}^{n_j} ((\partial_t u_\sigma^j, z_\sigma^j))_{j,l} + a(u_\sigma^j)(z_\sigma^j) + b(q_\sigma^j)(z_\sigma^j) + \sum_{l=0}^{n_j-1} ([u_\sigma^j]_l, z_{\sigma,l}^{j+}) + (u_{\sigma,0}^{j-} - s_\sigma^j, z_{\sigma,0}^{j-}) - ((f, z_\sigma^j))_j\}$$
$$+ \sum_{j=0}^{m-1} \{(s_h^{j+1} - u_\sigma^j(\tau_{j+1}), \lambda_h^{j+1})\} + (s_h^0 - u_0, \lambda_h^0)\Big\}.$$

We obtain by differentiation of the Lagrangian with respect to $\tilde{q}_\sigma, \tilde{u}_\sigma, \tilde{s}_h$ the fully discrete intervalwise optimality system as in (5.6) and the discrete matching conditions (5.8). This can either be verified as done for the continuous case by elementary calculus or by argumenting that discretization and dualization interchange for Galerkin discretizations. Now, the derivation of the error estimator proceeds as already seen for the classical DWR error estimator in Subsection 7.1. Let us state the following theorem:

Theorem 7.5. *(Error estimator for the multiple shooting solution) Let $(\tilde{q}_\sigma, \tilde{u}_\sigma, \tilde{s}_h)$ be a stationary point of the fully discretized problem* (5.6) *with matching conditions* (5.8)*. Then, for the functional error the following identity holds:*

$$\begin{aligned}
&\mathcal{J}(\tilde{q},\tilde{u},\tilde{s}) - \mathcal{J}(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h) \\
&\qquad = \frac{1}{2}\tilde{\mathcal{L}}'(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h,\tilde{z}_\sigma,\tilde{\lambda}_h)(\tilde{q}-\hat{\tilde{q}}_\sigma,\tilde{u}-\hat{\tilde{u}}_\sigma,\tilde{s}-\hat{\tilde{s}}_h,\tilde{z}-\hat{\tilde{z}}_\sigma,\tilde{\lambda}-\hat{\tilde{\lambda}}_h) + R_\sigma \\
&= \sum_{j=0}^{m-1}\Big\{\frac{1}{2}\{\tilde{\rho}_u^j(q_\sigma^j,u_\sigma^j,s_h^j)(z^j-\hat{z}_\sigma^j) + \tilde{\rho}_z^j(q_\sigma^j,u_\sigma^j,z_\sigma^j,\lambda_h^j)(u^j-\hat{u}_\sigma^j) + \tilde{\rho}_q^j(q_\sigma^j,u_\sigma^j,z_\sigma^j)(q^j-\hat{q}_\sigma^j)\} \\
&\qquad\qquad + \frac{1}{2}\{\tilde{\rho}_s^j(u_\sigma^j,s_h^j)(\lambda^j-\hat{\lambda}_h^j) + \tilde{\rho}_\lambda^j(z_\sigma^j,\lambda_h^j)(s^j-\hat{s}_h^j)\}\Big\} + R_\sigma \qquad (7.15)
\end{aligned}$$

for arbitrary $\hat{u}_\sigma^j, \hat{z}_\sigma^j \in \tilde{X}_{kh}^{j,r,s}$, $\hat{q}_\sigma^j \in Q_d^j$, $\hat{s}_h^j, \hat{\lambda}_h^j \in H_h^j$ and R_σ a remainder term of the same form as in Lemma 7.1 with $L = \tilde{\mathcal{L}}$. The residuals are defined as follows:

$$\begin{aligned}
\tilde{\rho}_u^j(q_\sigma^j,u_\sigma^j,s_h^j)(z^j-\hat{z}_\sigma^j) &:= \sum_{l=1}^{n_j}((\partial_t u_\sigma^j, z_\sigma^j-\hat{z}_\sigma^j))_{j,l} + a(u_\sigma^j)(z_\sigma^j-\hat{z}_\sigma^j) + b(q_\sigma^j)(z_\sigma^j-\hat{z}_\sigma^j) \\
&\quad + \sum_{l=0}^{n_j-1}([u_\sigma^j]_l, z_{\sigma,l}^{j+}-\hat{z}_{\sigma,l}^{j+}) + (u_{\sigma,0}^{j-}-s_h^j, z_{\sigma,0}^{j-}-\hat{z}_{\sigma,0}^{j-}) - ((f, z_\sigma^j-\hat{z}_\sigma^j))_j, \qquad (7.16a)
\end{aligned}$$

$$\begin{aligned}
\tilde{\rho}_z^j(q_\sigma^j,u_\sigma^j,z_\sigma^j,\lambda_h^j)(u^j-\hat{u}_\sigma^j) &:= -\alpha_1 J_1^{j\prime}(u_\sigma^j)(u_\sigma^j-\hat{u}_\sigma^j) \\
&\quad + \sum_{l=1}^{n_j}\Big\{-((\partial_t z_\sigma^j, u_\sigma^j-\hat{u}_\sigma^j))_{j,l}\Big\} + a_u'(u_\sigma^j)(u_\sigma^j-\hat{u}_\sigma^j, z_\sigma^j) \\
&\quad - \sum_{l=0}^{n_j-1}([z_\sigma^j]_l, u_{\sigma,l}^- - \hat{u}_{\sigma,l}^-)(z_{\sigma,n_j}^{j-} - \lambda_h^{j+1}, u_{\sigma,n_j}^- - \hat{u}_{\sigma,n_j}^-), \qquad (7.16b)
\end{aligned}$$

$$\tilde{\rho}_q^j(q_\sigma^j,u_\sigma^j,z_\sigma^j)(q^j-\hat{q}_\sigma^j) := \{b_q'(q_\sigma^j)(z_\sigma^j, q^j-\hat{q}_\sigma^j) - \alpha_3(q_\sigma^j, q^j-\hat{q}_\sigma^j)_Q, \qquad (7.16c)$$

$$\tilde{\rho}_s^j(u_\sigma^j,s_h^j)(\lambda^j-\hat{\lambda}_h^j) := \begin{cases} (s_h^j - u_0, \lambda^j-\hat{\lambda}_h^j) & j=0, \\ (s_h^j - u_\sigma^{j-1}(\tau_j), \lambda^j-\hat{\lambda}_h^j) & j=1,\dots,m, \end{cases} \qquad (7.17a)$$

$$\tilde{\rho}_\lambda^j(z_\sigma^j,\lambda_h^j)(s^j-\hat{s}_h^j) := \begin{cases} (\lambda^j - z_\sigma^j(\tau_j), s^j-\hat{s}_h^j) & j=0,\dots,m-1, \\ (\lambda^j, s^j-\hat{s}_h^j) - \alpha_2 J_2'(s^j)(s^j-\hat{s}_h^j) & j=m. \end{cases} \qquad (7.17b)$$

Proof. From the validity of the discretized formulation of condition (7.13b) and (7.13c) for both continuous and fully discrete stationary points of the problem, we obtain as for the classical error estimator

$$\mathcal{J}(\tilde{q},\tilde{u},\tilde{s}) - \mathcal{J}(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h) = \tilde{\mathcal{L}}(\tilde{q},\tilde{u},\tilde{s},\tilde{z},\tilde{\lambda}) - \tilde{\mathcal{L}}(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h,\tilde{z}_\sigma,\tilde{\lambda}_h). \tag{7.18}$$

With the same argumentation as in Lemma (7.2) we can conclude that the following equalities hold for all $(\tilde{\chi}_\sigma,\tilde{\varphi}_\sigma,\tilde{\xi}_h,\tilde{\psi}_\sigma,\tilde{\eta}_h) \in \tilde{Q}_d \times \tilde{X}_{kh} \times \tilde{H}_h \times \tilde{X}_{kh} \times \tilde{H}_h$:

$$\begin{aligned}\tilde{\mathcal{L}}'(\tilde{q},\tilde{u},\tilde{s},\tilde{z},\tilde{\lambda})(\tilde{\chi}_\sigma,\tilde{\varphi}_\sigma,\tilde{\xi}_h,\tilde{\psi}_\sigma,\tilde{\eta}_h) &= 0,\\ \tilde{\mathcal{L}}'(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h,\tilde{z}_\sigma,\tilde{\lambda}_h)(\tilde{\chi}_\sigma,\tilde{\varphi}_\sigma,\tilde{\xi}_h,\tilde{\psi}_\sigma,\tilde{\eta}_h) &= 0.\end{aligned}$$

We choose the spaces

$$\begin{aligned}Y_1 &:= \tilde{Q} \times \tilde{X} \times \tilde{H} \times \tilde{X} \times \tilde{H},\\ Y_2 &:= \tilde{Q}_d \times \tilde{X}_{kh} \times \tilde{H}_h \times \tilde{X}_{kh} \times \tilde{H}_h,\\ Y &:= Y_1 + Y_2\end{aligned}$$

and apply Lemma (7.1) to the right hand side of (7.18). This yields the relation

$$\mathcal{J}(\tilde{q},\tilde{u},\tilde{s}) - \mathcal{J}(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h) = \frac{1}{2}\tilde{\mathcal{L}}'(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h,\tilde{z}_\sigma,\tilde{\lambda}_h)(\tilde{q}-\hat{\tilde{q}}_\sigma,\tilde{u}-\hat{\tilde{u}}_\sigma,\tilde{s}-\hat{\tilde{s}}_h,\tilde{z}-\hat{\tilde{z}}_\sigma,\tilde{\lambda}-\hat{\tilde{\lambda}}_h) + R_\sigma.$$

Explicit calculation of $\tilde{\mathcal{L}}'$ directly yields the stated condition in terms of the residuals. □

A closer investigation of the error representation (7.15) reveals the composition of the error. First, the intervalwise discretization errors replay in the residuals $\tilde{\rho}_u^j$, $\tilde{\rho}_z^j$, and $\tilde{\rho}_q^j$ of the equations. Second, the contribution of the projection errors on the nodes is given by the residuals $\tilde{\rho}_s$ and $\tilde{\rho}_\lambda$ of the matching conditions.

Remark 7.6. (Simplifications) Let us note at this point that special choices of the spaces H_h^j for the shooting variables yield simplifications of the error estimator. First, if we choose H_h^j as the superset of the trace spaces from both adjacent intervals (equivalently, the mesh at τ_j is the common refinement of both adjacent meshes), then the shooting residuals (7.17) vanish, and the classical error estimator presented in the previous section remains.

Alternatively, we can shift the errors due to inconsistencies at the shooting nodes completely to the shooting residuals by choosing H_h^j as the intersection of the two adjacent trace spaces. In this case, the terms corresponding to the boundary values of primal and dual residual in (7.16) vanish. While this approach would yield the smallest shooting system, we did not use it here: it can be feared that it may yield an uncontrollable loss of accuracy at the nodes if the shooting residuals are not properly taken into account for the refinement of the adjacent meshes.

On the other hand, if H_h^j is equal to the trace space from the left or right, then the terms according to the right and the left boundary value in the second and first residual in (7.16) vanish, respectively. Additionally, either $\tilde{\rho}_s^j$ for $j = 1,\dots,m$ or $\tilde{\rho}_\lambda^j$ for $j = 0,\dots,m-1$ in (7.17) vanish as well.

We follow this last approach in the numerical experiments, where we choose H_h^j to be the trace space from the right hand side, yielding the simplified residuals

$$\tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j, s_h^j)(z^j - \hat{z}_\sigma^j) = \sum_{l=1}^{n_j} (\!(\partial_t u_\sigma^j, z_\sigma^j - \hat{z}_\sigma^j)\!)_{j,l} + a(u_\sigma^j)(z_\sigma^j - \hat{z}_\sigma^j) + b(q_\sigma^j)(z_\sigma^j - \hat{z}_\sigma^j) + \sum_{l=0}^{n_j-1} ([u_\sigma^j]_l, z_{\sigma,l}^{j+} - \hat{z}_{\sigma,l}^{j+}) - (\!(f, z_\sigma^j - \hat{z}_\sigma^j)\!)_j, \quad (7.19a)$$

$$\tilde{\rho}_\lambda^j(z_\sigma^j, \lambda_h^j)(s^j - \hat{s}_h^j) = \begin{cases} 0 & j = 0, \dots, m-1, \\ (\lambda^m, s^m - \hat{s}_h^m) - \alpha_2 J_2'(s^m)(s^m - \hat{s}_h^m) & j = m. \end{cases} \quad (7.19b)$$

All other residuals remain unchanged.

We have developed an error estimator for both cases of mesh handling, dynamically changing meshes with identical adjacent meshes on the nodes and intervalwise constant meshes with differently refined adjacent meshes. Consequently, we have to discuss how the error estimators can be evaluated in practice, that is how the exact solutions can be approximated properly for the evaluation of the error estimators.

Remark 7.7. In the introduction to this chapter we already mentioned, that we are only interested in the estimation of the spatial discretization error. Nevertheless, from the theoretical point of view, the whole discretization error can be treated analogously. However, the practical implementation of time-space finite elements is highly sophisticated and is not done in this thesis. Thus, in the following, we consider (q_k, u_k, z_k) and $(\hat{q}_k, \hat{u}_k, \hat{z}_k)$ instead of the continuous solution. This is allowed because all derivations done so far in this chapter hold for the time discrete formulation, too.

7.3 Evaluation of the Error Estimators

In this section, we present possible evaluation techniques for the a posteriori error estimators which we developed in the previous sections. Our main goal is the evaluation of the error estimators presented in Subsections 7.2 and 7.1. We want to evaluate the estimator for the discretization error such that the resulting error indicators reflect the error distribution in space properly, and the error estimator converges to the discretization error for $h \searrow 0$.

Remark 7.8. Throughout the remaining part of this chapter, we restrict ourselves to the two dimensional case of quadrilateral meshes. A generalization to the three dimensional case of hexalateral meshes is straightforward.

In the sequel, we want to consider the evaluation of the discretization error estimators presented before. First, for both error estimators, the evaluation procedure of the intervalwise residuals includes an approximation of the interpolation errors

$$(z_k^j - \hat{z}_\sigma^j), \quad (u_k^j - \hat{u}_\sigma^j), \quad (q_k^j - \hat{q}_\sigma^j),$$

where $\hat{z}_\sigma^j = I_h^{(s)} z_k^j \in \tilde{X}_{kh}^{j,r,s}$, $\hat{u}_\sigma^j = I_h^{(s)} u_k^j \in \tilde{X}_{kh}^{j,r,s}$ and $\hat{q}_\sigma^j = I_d q_k^j \in Q_d^j$ are the corresponding interpolants of the current solutions. Furthermore, for the evaluation of (7.15), the interpolation errors

$$(s^j - \hat{s}_h^j), \quad (\lambda^j - \hat{\lambda}_h^j)$$

with $\hat{s}_h^j = I_h^{(s)} s^j \in H_h^j$, $\hat{\lambda}_h^j = I_h^{(s)} \lambda^j \in H_h^j$ have to be approximated.

We introduce linear operators that map the computed solutions to approximations of these interpolation errors as follows:

$$\begin{aligned}
z_k^j - I_h^{(s)} z_k^j &\approx P_h z_\sigma^j, \\
u_k^j - I_h^{(s)} u_k^j &\approx P_h u_\sigma^j, \\
q_k^j - I_d q_k &\approx P_d q_\sigma^j, \\
s^j - I_h^{(s)} s^j &\approx \hat{P}_h \lambda_h^j, \\
\lambda^j - I_h^{(s)} \lambda^j &\approx \hat{P}_h s_h^j.
\end{aligned}$$

For the special case of quadrilateral meshes and a cG(s) space discretization, a concretization of the operators P_h and $\hat{P}_h$ is given as follows:

$$\begin{aligned}
P_h &= I_{2h}^{(2s)} - \mathrm{id} \quad \text{with} \qquad I_{2h}^{(2s)} : \tilde{X}_{kh}^{j,r,s} \to \tilde{X}_{k(2h)}^{j,r,2s}, \\
\hat{P}_h &= \hat{I}_{2h}^{(2s)} - \mathrm{id} \quad \text{with} \qquad \hat{I}_{2h}^{(2s)} : H_h^j \to H_{(2h)}^j.
\end{aligned}$$

Hereby, the piecewise biquadratic interpolation $I_{2h}^{(2s)}$ can easily be computed due to the requested patch structure of the refined mesh introduced in Section 5.2. The second operator P_d has to be defined according to the choice of Q_d. According to Section 5.4 we restrict ourselves to a cG(p)dG(r), $p \leq s$, discretization of the control space Q. Therefore, an appropriate choice of P_d is given by

$$P_h = I_{2h}^{(2p)} - \mathrm{id} \quad \text{with} \quad I_{2h}^{(2p)} : \tilde{X}_{kh}^{j,r,p} \to \tilde{X}_{k(2h)}^{j,r,2p}.$$

Remark 7.9. Whenever the discretization of the states and control coincide, that is $p = s$, the error due to the control equation vanishes:

$$\tilde{\rho}_q^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(q^j - \hat{q}_\sigma^j) = 0.$$

We discuss the practical evaluation of the error terms in the following. We easily obtain a fully computable version of both error estimators, (7.11) and (7.15), as follows: For the intervalwise formulation of the classical DWR error estimator, we obtain

$$\begin{aligned}
&J(q_k, u_k) - J(q_\sigma, u_\sigma) \\
&= \frac{1}{2}\Big\{ \sum_{j=0}^{m-1} \tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_h z_\sigma^j) + \sum_{j=0}^{m-1} \tilde{\rho}_z^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_h u_\sigma^j) + \sum_{j=0}^{m-1} \tilde{\rho}_q^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_d q_\sigma^j) \Big\} + R_\sigma.
\end{aligned}$$

Accordingly, the computable formulation of the error estimator for the complete multiple shooting system is given by

$$\begin{aligned}\mathcal{J}(\tilde{q}_k,\tilde{u}_k,\tilde{s}) - \mathcal{J}(\tilde{q}_\sigma,\tilde{u}_\sigma,\tilde{s}_h) \\ = \sum_{j=0}^{m-1} \Big\{ \frac{1}{2}\{\tilde{\rho}_u^j(q_\sigma^j,u_\sigma^j,s_h^j)(P_h z_\sigma^j) + \tilde{\rho}_z^j(q_\sigma^j,u_\sigma^j,z_\sigma^j,\lambda_h^j)(P_h u_\sigma^j) + \tilde{\rho}_q^j(q_\sigma^j,u_\sigma^j,z_\sigma^j)(P_d q_\sigma^j)\} \\ + \frac{1}{2}\{\tilde{\rho}_s^j(u_\sigma^j,s_h^j)(\hat{P}_h\lambda_h^j) + \tilde{\rho}_\lambda^j(z_\sigma^j,\lambda_h^j)(\hat{P}_h s_h^j)\}\Big\} + R_\sigma.\end{aligned}$$

Finally, to conclude this chapter, let us present some numerical examples in order to verify the validity of the error estimators.

7.4 Numerical Examples

We present numerical results of the error estimators for Example 2.1 and 2.4. As mentioned before, the error estimator for the whole multiple shooting system with intervalwise constant meshes is considered to be more promising from the computational point of view. Therefore, we mainly focus on this approach henceforth, but nevertheless give comparative results on the classical DWR error estimator for dynamically changing meshes, too. We start with an example constrained by the heat equation with control as a source term on the right-hand side. The cost functional measures the terminal time deviation from a desired stated.

Example 7.2. (Distributed control of the heat equation) Considering Example 2.1 with parameters $\alpha_2 = 1$, $\alpha_3 = 10^{-3}$, $T = 2.5$, and desired terminal time state $\bar{u} = 0.5$, we apply multiple shooting with a total number of 10 intervals. The intervals are subdivided into 25 time steps each, that is a constant time step size of 0.01 for the time discretization. The spatial domain is chosen as a square $\Omega = (-1,1)\times(-1,1)$. Furthermore, the initial condition is given by the function $u(0) = \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}x)$, and the boundary conditions are given as homogenous Dirichlet conditions.

We consider global refinement after convergence of the multiple shooting method and restart the multiple shooting method on the new mesh. This yields the convergence results for the multiple shooting error estimator as shown in Table 7.1. The numerical computations illustrate that with increasing refinement level, the error estimator converges to the correct value of the functional error. In this case of identical, globally refined meshes on all intervals, the projection error is due to the projection of u_0 into the discretized space. We also consider the error estimator for the case of locally refined meshes. Therefore, we apply the mesh refinement techniques which will be introduced in Chapter 8. As expected, for locally refined, intervalwise constant meshes the results are accordingly, as outlined in Table 7.2. In the following, N is the total number of cells from all meshes (of all time steps), e_h denotes the correct error value, η_{disc} denotes the estimate of the discretization error, η_{proj} is the estimate of the projection error, and η is the estimate for the whole functional error e_h. The so called efficiency index $I_{\text{eff}} := {}^{\eta_h}/_{\text{e}_h}$ gives information on the accuracy of the error estimator and on the development of the approximation. I_{eff} should converge to 1 for $N \to \infty$ if the error estimator works well.

Table 7.1: Errors and estimates for global refinement with intervalwise constant meshes for Example 7.2 with functional value $J(q,u) = 0.0553066$.

N	e_h	η_{disc}	η_{proj}	η	I_{eff}
1080	$-1.669 \cdot 10^{-01}$	$-6.090 \cdot 10^{-02}$	$-6.167 \cdot 10^{-11}$	$-6.090 \cdot 10^{-02}$	0.365
4320	$-8.395 \cdot 10^{-02}$	$-2.006 \cdot 10^{-02}$	$-1.384 \cdot 10^{-11}$	$-2.006 \cdot 10^{-02}$	0.239
17280	$-2.824 \cdot 10^{-02}$	$-9.728 \cdot 10^{-03}$	$-1.152 \cdot 10^{-12}$	$-9.728 \cdot 10^{-03}$	0.344
69120	$-7.162 \cdot 10^{-03}$	$-5.519 \cdot 10^{-03}$	$6.500 \cdot 10^{-12}$	$-5.519 \cdot 10^{-03}$	0.770
276480	$-1.718 \cdot 10^{-03}$	$-1.933 \cdot 10^{-03}$	$1.396 \cdot 10^{-13}$	$-1.933 \cdot 10^{-03}$	1.125
1105920	$-4.225 \cdot 10^{-04}$	$-4.522 \cdot 10^{-04}$	$8.020 \cdot 10^{-14}$	$-4.522 \cdot 10^{-04}$	1.070
4423680	$-1.050 \cdot 10^{-04}$	$-1.073 \cdot 10^{-04}$	$1.981 \cdot 10^{-15}$	$-1.073 \cdot 10^{-04}$	1.022

Table 7.2: Errors and estimates for local refinement with intervalwise constant meshes for Example 7.2 with functional value $J(q,u) = 0.0553066$.

N	e_h	η_{disc}	η_{proj}	η	I_{eff}
1080	$-1.669 \cdot 10^{-01}$	$-6.090 \cdot 10^{-02}$	$6.166 \cdot 10^{-11}$	$-6.090 \cdot 10^{-02}$	0.365
2052	$-8.395 \cdot 10^{-02}$	$-2.006 \cdot 10^{-02}$	$-1.020 \cdot 10^{-07}$	$-2.006 \cdot 10^{-02}$	0.239
4644	$-2.824 \cdot 10^{-02}$	$-9.729 \cdot 10^{-03}$	$-1.714 \cdot 10^{-07}$	$-9.729 \cdot 10^{-03}$	0.344
8532	$-7.165 \cdot 10^{-03}$	$-5.521 \cdot 10^{-03}$	$-1.912 \cdot 10^{-07}$	$-5.521 \cdot 10^{-03}$	0.771
17604	$-1.721 \cdot 10^{-03}$	$-1.939 \cdot 10^{-03}$	$-1.916 \cdot 10^{-07}$	$-1.940 \cdot 10^{-03}$	1.127
37044	$-4.161 \cdot 10^{-04}$	$-4.323 \cdot 10^{-04}$	$-1.917 \cdot 10^{-07}$	$-4.325 \cdot 10^{-04}$	1.039
77220	$-1.809 \cdot 10^{-04}$	$-1.631 \cdot 10^{-04}$	$-1.917 \cdot 10^{-07}$	$-1.633 \cdot 10^{-04}$	0.903
139428	$-1.079 \cdot 10^{-04}$	$-1.092 \cdot 10^{-04}$	$-1.974 \cdot 10^{-07}$	$-1.094 \cdot 10^{-04}$	1.013
291060	$-5.213 \cdot 10^{-05}$	$-5.095 \cdot 10^{-05}$	$-1.974 \cdot 10^{-07}$	$-5.115 \cdot 10^{-05}$	0.981

Local refinement with dynamically changing meshes yields different meshes within the process of mesh adaptation. However, the error estimator η in this case converges to the correct value e_h as given in Table 7.3.

Table 7.3: Errors and estimates for local refinement with dynamically changing meshes for Example 7.2 with functional value $J(q,u) = 0.0553066$.

N	e_h	η	I_{eff}
1080	$-1.669 \cdot 10^{-01}$	$-6.090 \cdot 10^{-02}$	0.365
1956	$-8.395 \cdot 10^{-02}$	$-2.006 \cdot 10^{-02}$	0.239
4152	$-2.824 \cdot 10^{-02}$	$-9.730 \cdot 10^{-03}$	0.345
8592	$-7.168 \cdot 10^{-03}$	$-5.524 \cdot 10^{-03}$	0.771
17412	$-1.724 \cdot 10^{-03}$	$-1.938 \cdot 10^{-03}$	1.124
35796	$-4.282 \cdot 10^{-04}$	$-4.575 \cdot 10^{-04}$	1.068
75612	$-1.094 \cdot 10^{-04}$	$-1.114 \cdot 10^{-04}$	1.018
165804	$-2.966 \cdot 10^{-05}$	$-2.981 \cdot 10^{-05}$	1.005

As expected, by refinement with dynamically changing meshes we obtain better approximations of the functional at a smaller number of cells. We hint at the fact, that by choosing a sufficiently large number of intervals, we can compensate this drawback for intervalwise constant meshes.

To complete this section of numerical examples, we consider now an example constrained by a nonlinear PDE with Neumann boundary control. We minimize a time distributed cost functional and thereby approximate a time distributed function $\bar{u}(t)$.

Example 7.3. (Neumann boundary control) We consider Example 2.4 on a spatial domain $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$ with initial condition $u_0 = 1$ and desired time distributed state $\bar{u}(t) = t \cdot \cos(4\pi x)\cos(4\pi y)$. For the regularization parameter we chose $\alpha_3 = 10^{-4}$ and for the time domain $I = (0, 2)$. The number of multiple shooting intervals is chosen as $m = 10$, and the time stepping scheme is initialized with a constant time step size of 0.01. Furthermore, the intervalwise controls are requested to be constant in time, that is for $j = 0, \ldots, 10$

$$Q^j = \left\{ v \in L^2(I_j, R) \;\middle|\; v(t) = c_j \in R \right\} \subset L^2(I_j, R).$$

As in the previous example, the results for Example 7.3 show an appropriate convergence behavior of the error estimator for the distributed cost functional.

In this example, we consider only the case of intervalwise constant meshes. In Table 7.4 the obtained error and the corresponding error estimator for the different steps of global mesh refinement are presented.

For the sake of completeness Table 7.5 replays the analogous results for local refinement with intervalwise constant meshes. In both cases the error estimator η converges to the correct value e_h with proceeding mesh refinement, and the efficiency index I_{eff} converges accordingly to 1.

Table 7.4: Errors and estimates for global refinement for Example 7.3 with functional value $J(q, u) = 0.295083$.

N	e_h	η_{disc}	η_{proj}	η	I_{eff}
3360	$2.299 \cdot 10^{-02}$	$2.020 \cdot 10^{-02}$	0	$-2.020 \cdot 10^{-02}$	-0.879
13440	$8.932 \cdot 10^{-03}$	$-4.174 \cdot 10^{-04}$	0	$4.174 \cdot 10^{-04}$	0.047
53760	$2.314 \cdot 10^{-03}$	$-2.184 \cdot 10^{-03}$	0	$2.184 \cdot 10^{-03}$	0.944
215040	$5.709 \cdot 10^{-04}$	$-5.787 \cdot 10^{-04}$	0	$5.787 \cdot 10^{-04}$	1.014
860160	$1.420 \cdot 10^{-04}$	$-1.427 \cdot 10^{-04}$	0	$1.427 \cdot 10^{-04}$	1.005
3440640	$3.534 \cdot 10^{-05}$	$-3.548 \cdot 10^{-05}$	0	$3.548 \cdot 10^{-05}$	1.004

We have seen so far that the error estimators yield reliable estimates for the total functional error. In addition to the correct approximation of the error, we are interested in appropriate adaptive mesh refinement. The results presented above and below already indicate that local mesh refinement for these examples yields better approximations of the functional error at fewer cells than global mesh refinement. In general, we want to refine those cells with large error contributions. Therefore, we need knowledge on the cellwise error distribution.

Table 7.5: Errors and estimates for local refinement with intervalwise constant meshes for Example 7.3 with functional value $J(q,u) = 0.295083$.

N	e_h	η_{disc}	η_{proj}	η	I_{eff}
3360	$2.299 \cdot 10^{-02}$	$2.020 \cdot 10^{-02}$	$-0.000 \cdot 10^{-00}$	$-2.020 \cdot 10^{-02}$	-0.879
7644	$1.154 \cdot 10^{-02}$	$3.663 \cdot 10^{-03}$	$-1.735 \cdot 10^{-05}$	$-3.661 \cdot 10^{-03}$	-0.317
18480	$6.665 \cdot 10^{-03}$	$-1.179 \cdot 10^{-03}$	$3.009 \cdot 10^{-05}$	$1.176 \cdot 10^{-03}$	0.176
43176	$2.339 \cdot 10^{-03}$	$-1.637 \cdot 10^{-03}$	$1.965 \cdot 10^{-07}$	$1.637 \cdot 10^{-03}$	0.700
98112	$8.231 \cdot 10^{-04}$	$-6.540 \cdot 10^{-04}$	$-4.217 \cdot 10^{-08}$	$6.541 \cdot 10^{-04}$	0.795
225372	$3.260 \cdot 10^{-04}$	$-3.163 \cdot 10^{-04}$	$-5.213 \cdot 10^{-09}$	$3.163 \cdot 10^{-04}$	0.970
518448	$1.385 \cdot 10^{-04}$	$-1.395 \cdot 10^{-04}$	$-1.876 \cdot 10^{-09}$	$1.395 \cdot 10^{-04}$	1.007
1063776	$5.960 \cdot 10^{-05}$	$-5.963 \cdot 10^{-05}$	$-4.521 \cdot 10^{-10}$	$5.963 \cdot 10^{-05}$	1.000
2116128	$3.095 \cdot 10^{-05}$	$-3.120 \cdot 10^{-05}$	$-5.949 \cdot 10^{-11}$	$3.120 \cdot 10^{-05}$	1.008

The next chapter is first about the localization of the discretization error and second about the process of mesh adaptation within the multiple shooting method.

8 Multiple Shooting and Mesh Adaptation

In this chapter, we present *localization techniques* for the error estimators developed in Chapter 7 and an *adaptive algorithm* for the purpose of mesh adaptation. Mesh adaptation plays an important role in the context of efficient solution techniques for PDEs and aims at the determination of an optimal mesh in time and space with respect to the desired accuracy. That is, we search for a preferably coarse mesh on which the solution is approximated such that we reach a beforehand defined error bound for the cost functional.

We subdivide this chapter into two main parts. Section 8.1 is on the mesh adaptation for the classical DWR error estimator in general. It is subdivided into Subsection 8.1.1 on the details of the localization procedure for the error and Subsection 8.1.2 on the mesh adaptation algorithm. Section 8.2 is engaged with the corresponding procedures for the newly developed error estimator for the multiple shooting system from Chapter 7. It is analogously partitioned into Subsections 8.2.1 and 8.2.2. The chapter is closed by substantiating the obtained algorithms by numerical examples in Section 8.3.

8.1 Mesh Adaptation by the Classical DWR Error Estimator

8.1.1 Localization of the Error Estimator

We start this section by introducing some notation. The error estimators are generally denoted by η with appropriate indices for the specification in the particular context:

$$\eta_h := \frac{1}{2}\left\{\tilde{\rho}_u(q_\sigma, u_\sigma)(P_h z_\sigma) + \tilde{\rho}_z(q_\sigma, u_\sigma, z_\sigma)(P_h u_\sigma) + \tilde{\rho}_q(q_\sigma, u_\sigma, z_\sigma)(P_d q_\sigma)\right\},$$
$$\eta_h^j := \frac{1}{2}\left\{\tilde{\rho}_u^j(q_\sigma^j, u_\sigma^j)(P_h z_\sigma^j) + \tilde{\rho}_z^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_h u_\sigma^j) + \tilde{\rho}_q^j(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_d q_\sigma^j)\right\},$$

and thus

$$\eta_h = \sum_{j=0}^{m-1} \eta_h^j.$$

Furthermore the intervalwise error estimator η_h^j is split into its subintervalwise contributions,

$$\eta_h^j = \sum_{l=1}^{n_j} \eta_{h,l}^j,$$

with

$$\eta_{h,l}^j = \frac{1}{2}\left\{\tilde{\rho}_u^{j,l}(q_\sigma^j, u_\sigma^j)(P_h z_\sigma^j) + \tilde{\rho}_z^{j,l}(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_h u_\sigma^j) + \tilde{\rho}_q^{j,l}(q_\sigma^j, u_\sigma^j, z_\sigma^j)(P_d q_\sigma^j)\right\}.$$

Here $\tilde{\rho}_u^{j,l}$, $\tilde{\rho}_z^{j,l}$ and $\tilde{\rho}_q^{j,l}$ denote those parts of the residuals $\tilde{\rho}_u^j$, $\tilde{\rho}_z^j$ and $\tilde{\rho}_q^j$ which belong to the time subinterval $I_{j,l}$.

With this notation at hand we proceed with the meshwise localization procedure. We want to break up the error contributions $\eta_{h,l}^j$ into *cellwise error indicators* which correspond to the cellwise contribution to the error of primal, dual, and control equation. For the latter, the splitting can be performed rather simple by writing the scalar products within the residual $\tilde{\rho}_q^{j,l}$ as a sum over the cells of the triangulation $\mathcal{T}_h^{j,l}$. Concerning the splitting of the primal and dual residuals, our approach follows the one presented in [7]. We transform the residuals $\tilde{\rho}_u^{j,l}$ and $\tilde{\rho}_z^{j,l}$ by cellwise partial integration of $(\nabla u_\sigma^j, \nabla P_h z_\sigma^j)$ and $(\nabla z_\sigma^j, \nabla P_h u_\sigma^j)$ such that we obtain

$$(\nabla u_\sigma^j, \nabla P_h z_\sigma^j) = \sum_{K \in \mathcal{T}_h^{j,l}} \left\{ (-\Delta u_\sigma^j, P_h z_\sigma^j)_K + (\partial_n u_\sigma^j, P_h z_\sigma^j)_{\partial K} \right\},$$
$$(\nabla z_\sigma^j, \nabla P_h u_\sigma^j) = \sum_{K \in \mathcal{T}_h^{j,l}} \left\{ (-\Delta z_\sigma^j, P_h u_\sigma^j) + (\partial_n z_\sigma^j, P_h u_\sigma^j)_{\partial K} \right\}.$$

We follow the arguments of Becker and Rannacher in [7] that $P_h u_\sigma^j$ and $P_h z_\sigma^j$ have continuous traces along the boundary ∂K of each cell K. Each inner face occurs twice within the sum with changing sign of the normal derivative (c.f. Figure 8.1). Therefore, we obtain

$$(\nabla u_\sigma^j, \nabla P_h z_\sigma^j) = \sum_{K \in \mathcal{T}_h^{j,l}} \left\{ (-\Delta u_\sigma^j, P_h z_\sigma^j)_K + \frac{1}{2}([\partial_n u_\sigma^j], P_h z_\sigma^j)_{\partial K} \right\},$$
$$(\nabla z_\sigma^j, \nabla P_h u_\sigma^j) = \sum_{K \in \mathcal{T}_h^{j,l}} \left\{ (-\Delta z_\sigma^j, P_h u_\sigma^j) + \frac{1}{2}([\partial_n z_\sigma^j], P_h u_\sigma^j)_{\partial K} \right\}.$$

Here, we define $[\partial_n u_\sigma^j]$ on inner faces as the jump over the face Γ from K to the neighbor cell K' and on outer faces as the outer normal derivative:

$$[\partial_n u_\sigma^j]\Big|_\Gamma := \partial_n u_\sigma^j\Big|_K + \partial_{n'} u_\sigma^j\Big|_{K'} \quad \text{and} \quad [\partial_n u_\sigma^j]\Big|_\Gamma := 2\partial_n u_\sigma^j.$$

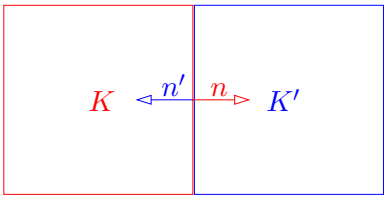


Figure 8.1: Adjacent cells and outer normal vector.

With the residuals transformed as described above and written as a sum over the cells of $\mathcal{T}_h^{j,l}$, we are now able to consider the cellwise contribution $\eta_{h,l,K}^j$ of the cell K to the residuals. Therefore, we write

$$\eta_{h,l}^j = \sum_{K \in \mathcal{T}_h^{j,l}} \eta_{h,l,K}^j \quad \text{and} \quad \eta_h = \sum_{j=0}^{m-1} \sum_{l=1}^{n_j} \sum_{K \in \mathcal{T}_h^{j,l}} \eta_{h,l,K}^j.$$

Next, we want to discuss the process of mesh adaptation according to the cellwise error indicators.

8.1.2 The Process of Mesh Adaptation

Whenever we decide to refine the spatial meshes after the evaluation of the error estimators, we have to assign a criterion that allows us to distinguish between those cells, which will be refined and those cells that will be kept the same. We wish to refine the mesh, such that the error contributions from different cells are equilibrated,

$$\eta^{j}_{h,l,K} \approx \text{const},$$

for all intervals I_j for $j = 0, 1, \ldots, m-1$, all subintervals $I_{j,l}$ for $l = 1, 2, \ldots, n_j$, and all cells $K \in \mathcal{T}_h^{j,l}$. For the sake of simplicity with respect to the notation, let us assume that we have a total number of M cells $K_1, \ldots, K_M$ with corresponding error indicators η^{K_n}, $n = 1, \ldots, M$. The process of equilibration is performed as follows. We sort the cell indicators in descending order,

$$\eta^{K_{\pi_1}} \geq \eta^{K_{\pi_2}} \geq \ldots \geq \eta^{K_{\pi_M}},$$

where $(\pi_1, \ldots \pi_M)$ denotes a permutation of $(1, \ldots, M)$. Then we choose a fixed number n_{ref} which denotes the fractional amount of cells which we wish to refine. Finally, we refine this fractional amount of cells which have the largest error contributions. In algorithmical formulation this procedure reads:

Algorithm 8.1 Spatial mesh adaptation for the classical DWR error estimator

Require: Set of initial meshes $\mathcal{T}_h^{j,l}$.

```
Choose n_ref.
Evaluate solutions q_σ, u_σ, z_σ .
Evaluate error estimators and retrieve cellwise indicators η^{K_i}.
Sort error indicators: η^{K_π1} ≥ η^{K_π2} ≥ ... ≥ η^{K_πM}.
for n = 1 to M do
    if n < M · n_ref then
        Set refinement indicator on K_πn.
Refine meshes according to the cellwise refinement indicators.
Replace adjacent meshes on the nodes by common refinement.
```

The procedure presented above describes only one cycle of spatial mesh adaptation. Usually, mesh adaptation is performed after each solution cycle of the optimality system. In the case of multiple shooting, this means the evaluation of the error estimator and the refinement of the cells after convergence of the outer Newton method. Furthermore, as already seen in the previous chapter, the approach of the classical DWR error estimator is limited by the assumption of equivalent adjacent meshes on the nodes. Therefore, a different refinement strategy is applied in the case of the error estimator developed in Section 7.2.

8.2 Mesh Adaptation by the Error Estimator for the Multiple Shooting System

Before we begin with the description of the adaptation strategy, let us state a preliminary assumption, namely, according to Chapter 5, we assume constant meshes on each of the multiple shooting intervals.

8.2.1 Localization of the Error Estimator

The localization procedure for the error estimator is quite similar to the one presented in the previous section. The error estimator for the multiple shooting system consists of two different types of intervalwise residuals. First, we have the intervalwise residual terms due to the discretization error on the intervals $\tilde{\rho}_u^j$, $\tilde{\rho}_z^j$, and $\tilde{\rho}_q^j$. Second, we have the residuals due to the projection error on the nodes $\tilde{\rho}_s^j$ and $\tilde{\rho}_\lambda^j$. We use the same abbreviations η_h^j and $\eta_{h,l}^j$ as before for the discretization error estimates and additionally introduce an abbreviation for the projection errors

$$\eta_{p_u}^j := \frac{1}{2}\tilde{\rho}_s^j(u_\sigma^j, s_h^j)(\hat{P}_h\lambda_h^j),$$
$$\eta_{p_z}^j := \frac{1}{2}\tilde{\rho}_\lambda^j(z_\sigma^j, \lambda_h^j)(\hat{P}_h s_h^j).$$

The localization procedure for η_h^j directly follows the approach for the DWR error estimator. Integrating back by parts, and summing up over all cells, we obtain as before

$$\eta_h^j = \sum_{l=1}^{n_j} \sum_{K\in\mathcal{T}_h^j} \eta_{h,l,K}^j.$$

The projection errors are easily localized to cellwise contributions $\eta_{p_u,K}^j$ and $\eta_{p_z,K}^j$ on the node mesh $\mathcal{T}_h^{\tau_j}$ by

$$\eta_{p_u}^j = \sum_{K\in\mathcal{T}_h^{\tau_j}} \eta_{p_u,K}^j \quad \text{and} \quad \eta_{p_z}^j = \sum_{K\in\mathcal{T}_h^{\tau_j}} \eta_{p_z,K}^j.$$

Over all, we obtain

$$\eta_h = \sum_{j=0}^{m-1}\sum_{l=1}^{n_j}\sum_{K\in\mathcal{T}_h^j} \eta_{h,l,K}^j + \sum_{j=0}^{m-1}\sum_{K\in\mathcal{T}_h^{\tau_j}} \left\{\eta_{p_u,K}^j + \eta_{p_z,K}^j\right\}.$$

Henceforth, considering the adaptation strategy, we have to define the choice of $\mathcal{T}_h^{\tau_j}$. As mentioned in Chapter 7, we follow the approach which is used in the numerical experiments. We have $\mathcal{T}_h^{\tau_j} = \mathcal{T}_h^j$, that is, the node mesh on τ_j is equivalent to the trace mesh from the right. Thus, according to (7.19), $\tilde{\rho}_\lambda^j(z_\sigma^j, \lambda_h^j)(\hat{P}_h s_h^j)$ vanishes, and only the projection error for the dual variable remains.

8.2.2 The Process of Mesh Adaptation

In contrast to dynamically changing meshes, intervalwise constant meshes need the calculation of appropriate cellwise error indicators not only out of the cellwise indicators in each time step but also out of the additional projection errors. Our basic idea is to calculate the cell indicator for a certain cell as the maximum absolute value over all time steps of the interval and the projection error. Let the cell indicator for a cell K of $\mathcal{T}_h^j$ be denoted by η_K^j, then

$$\eta_K^j := \max\left\{|\eta_{p_z,K}^j|, \max_{l=1,\dots,n_j} |\eta_{h,l,K}^j|\right\}. \tag{8.1}$$

As before we assume that we have a total number of N cells from the intervalwise meshes, $K_1, \dots, K_N$ for which we calculate the final error indicators η^{K_i} according to (8.1). In this notation, the adaptive algorithm reads:

Algorithm 8.2 Spatial mesh adaptation for the multiple shooting error estimator

Require: Set of initial meshes $\mathcal{T}_h^j$.
1: Chose n_{ref}.
2: Evaluate solutions $q_\sigma, u_\sigma, z_\sigma$.
3: Evaluate error estimators and cellwise indicators for all time steps and projection errors.
4: Calculate cellwise indicators η^{K_i}.
5: Sort these indicators: $\eta^{K_{\pi_1}} \geq \eta^{K_{\pi_2}} \geq \dots \geq \eta^{K_{\pi_M}}$.
6: **for** $n = 1$ **to** M **do**
7: **if** $n < M \cdot n_{\text{ref}}$ **then**
8: Set refinement indicator on K^{π_n}.
9: Refine meshes according to the cellwise refinement indicators.

We close this chapter by a consideration of two different example problems in the next section.

8.3 Numerical Examples

In this section, numerical examples with different types of cost functionals and controls are investigated.

Remark 8.1. From the viewpoint of multiple shooting in the context of partial differential equations, it is advisable to allow different meshes for states and control. This approach consequently allows the reduction of the dimension of the control space, which is of crucial importance for the efficiency of the multiple shooting method as outlined in Chapter 6. Due to implementational aspects, we are not yet able to consider this case of mesh refinement, such that in the following examples the meshes for states and control are equivalent.

First of all, we recall the formulation from Example 2.2 and consider two different time domain decompositions in the following. This example serves well to demonstrate the properties of local mesh refinement due to the chosen terminal time cost functional. For the resolution of the singularity on the boundary at the terminal time point we can expect a thin boundary layer of locally refined cells at the corresponding spatial mesh.

Example 8.1. (Distributed control of the heat equation) We choose the spatial domain $\Omega = (-1,1) \times (-1,1)$, the time domain $I = (0, 2.5)$, and the parameter $\alpha = 10^{-3}$. The optimal control problem of interest is given by

$$\min_{(q,u)\in Q\times X} J(q,u) := \frac{1}{2}\|u(T) - 0.5\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\int_I \|q(t)\|^2_{L^2(\Omega)} dt \tag{8.2a}$$

subject to the linear heat equation

$$\begin{aligned} \partial_t u - \Delta u &= q && \text{in} && \Omega \times I, \\ u &= 0 && \text{on} && \partial\Omega \times I, \\ u &= \cos\left(\frac{\pi}{2}x\right)\cos\left(\frac{\pi}{2}x\right) && \text{in} && \Omega \times \{0\}\,. \end{aligned} \tag{8.2b}$$

We want to minimize the terminal time functional constrained by the linear heat equation for which the control acts as a source term on the right-hand side. The need for mesh adaptation results from the choice of the goal, $J_1(q) = \frac{1}{2}\int_I \|u(T) - 0.5\|^2 dt$ with inhomogeneous boundary values while the solution is subject to a homogeneous Dirichlet boundary condition. We choose a constant time step size of 0.01 and extrapolate the correct value of the functional as 0.0553066. The results of the computations with local mesh refinement for 10 intervals are shown in Figure 8.2.

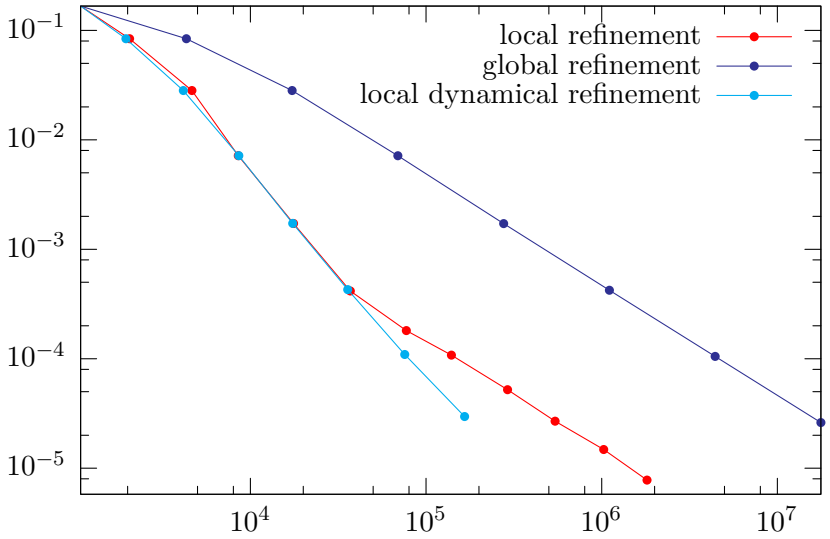


Figure 8.2: Functional error for differently fine meshes, gained by different strategies of mesh refinement for Example 8.1. (x-axis: $\log(N)$, y-axis: $\log(|e_h|)$)

Additionally, Figure 8.3 illustrates the relation between the projection error estimate and the error estimate for the discretization error on the intervals. Mesh refinement is performed appropriately such that the projection error estimate does not become dominant. It stays below the discretization error and decreases if necessary.

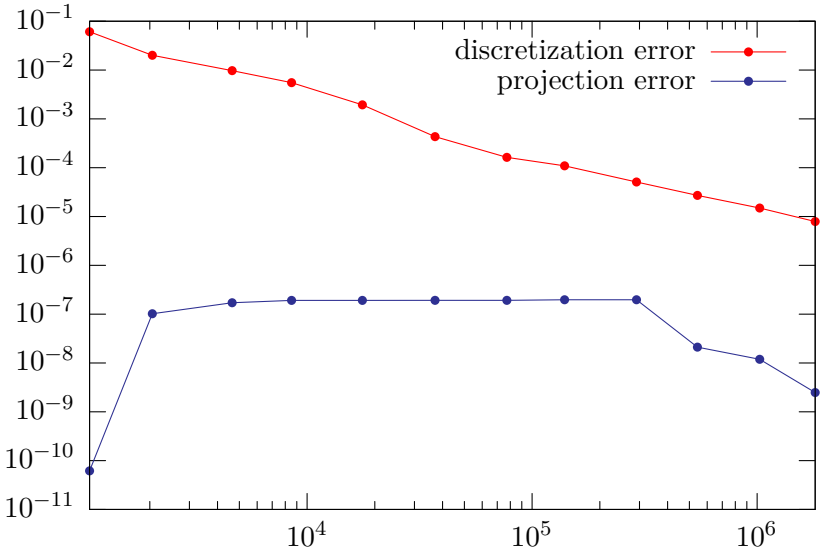


Figure 8.3: Discretization error estimate and projection error estimate with proceeding mesh refinement for Example 8.1. (x-axis: $\log(N)$, y-axis: $\log(|\eta_{\text{proj}}|)$ and $\log(|\eta_{\text{disc}}|)$)

We see that local refinement due to both dynamically changing and intervalwise constant meshes is by far better than global refinement. We need a larger number of cells for intervalwise constant meshes to obtain a certain error bound than for dynamically changing ones. This can be expected due to the less flexible spatial meshes, but nevertheless, this drawback can easily be overcome by increasing the number of intervals.

To demonstrate this property, we recalculate Example 8.1 for 50 intervals. The results presented in Figure 8.4 point out that a larger number of intervals yields significantly better results, in this case even better than with dynamically changing meshes. However, we remark that for further refinement, the curves will cross and refinement with dynamically changing meshes yields better results than with intervalwise constant meshes. The obtained sequence of locally refined meshes (see Figure 8.8 on page 128) underlines the assumption that refinement is particularly important for the resolution of the singularity on the boundary at the terminal time point.

In the following, we do not pursue the idea of dynamically changing meshes further on, but restrict ourselves on the application of intervalwise constant meshes. In the following example, we consider an optimization problem on a T-shaped domain (see Figure 8.5). We chose Neumann boundary control and a cost functional distributed in time and space. In this particular context of mesh adaptation, the choice of the domain, the control and the cost functional suggests the assumption that refinement should occur, first at the control boundary, and second at the inwards pointing corners. This is due to the control localized on the boundary of the lower part of the T-shaped domain which must affect the solution on the whole domain. Therefore, information must pass along the inwards pointing corners to control the solution in the upper part of the T-Shaped domain.

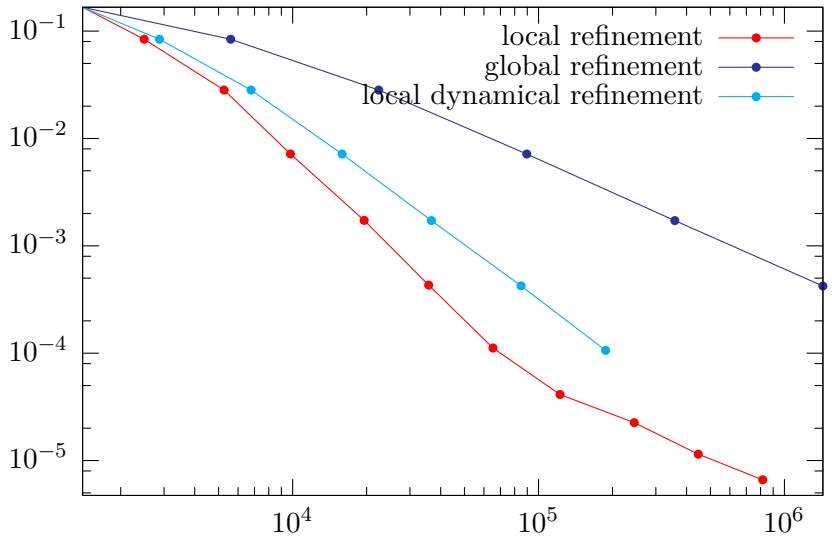


Figure 8.4: Functional error for differently fine meshes, gained by different strategies of mesh refinement (calculated on 50 intervals for Example 8.1). (x-axis: $\log(N)$, y-axis: $\log(|e_h|)$)

Example 8.2. (Neumann control on a T-shaped domain) We consider the problem

$$\min_{(q,u)\in Q\times X} J(q,u) = \frac{1}{2}\int_I \|u(t)-\bar{u}(t)\|^2 dt + \frac{\alpha}{2}\int_I \|q(t)\|_Q^2 dt$$

subject to the constraint

$$\begin{aligned} \partial_t u - \Delta u + u^3 - u = 0 \quad &\text{in} \quad I\times\Omega_T, \\ u(0) = 0 \quad &\text{in} \quad \Omega_T, \\ \partial_n u = \begin{cases} q & \text{on} \quad I\times\Gamma_c, \\ 0 & \text{on} \quad I\times\Omega_T\backslash\Gamma_c \end{cases} & \end{aligned}$$

with Ω_T a T-shaped domain (see Figure 8.5), $u_0 = 0$, $\bar{u} = t$, $\alpha = 10^{-3}$, and $I = (0,1)$.

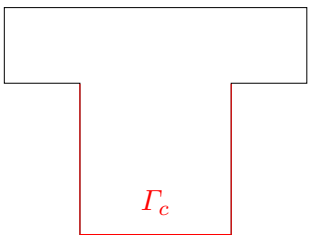


Figure 8.5: Ω_T with control boundary Γ_c.

We consider the case of 10 intervals, the implicit Euler with step size 0.01, and obtain by extrapolation the correct value of the functional as 0.0052103. In Figure 8.6 we compare global and local refinement for intervalwise constant meshes.

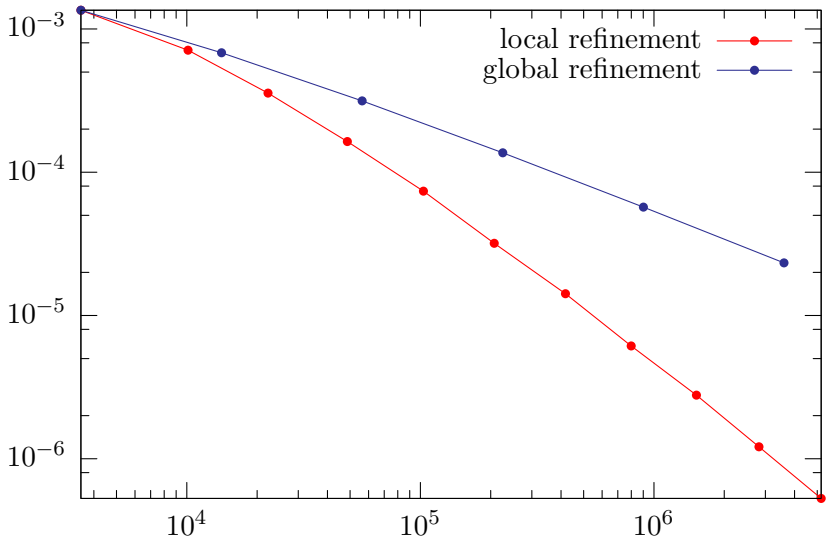


Figure 8.6: Functional error for differently fine meshes, gained by different strategies of mesh refinement for Example 8.2. (x-axis: $\log(N)$, y-axis: $\log(|e_h|)$)

The outstanding performance of local refinement is illustrated by the steep descent of the error value for the corresponding curve. The development of the different parts of the error estimator is given in Figure 8.7. It shows even more clearly than in the example before, that the estimate for the projection error decreases accordingly with the estimate for the discretization error.

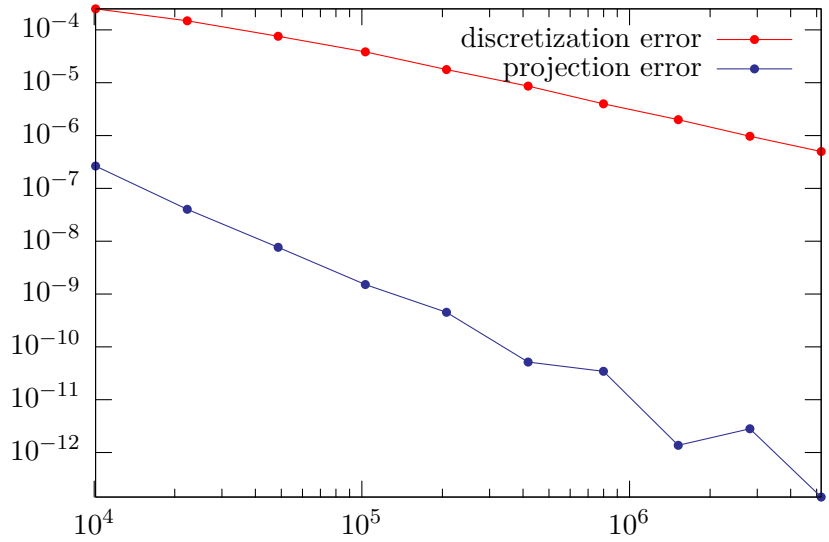
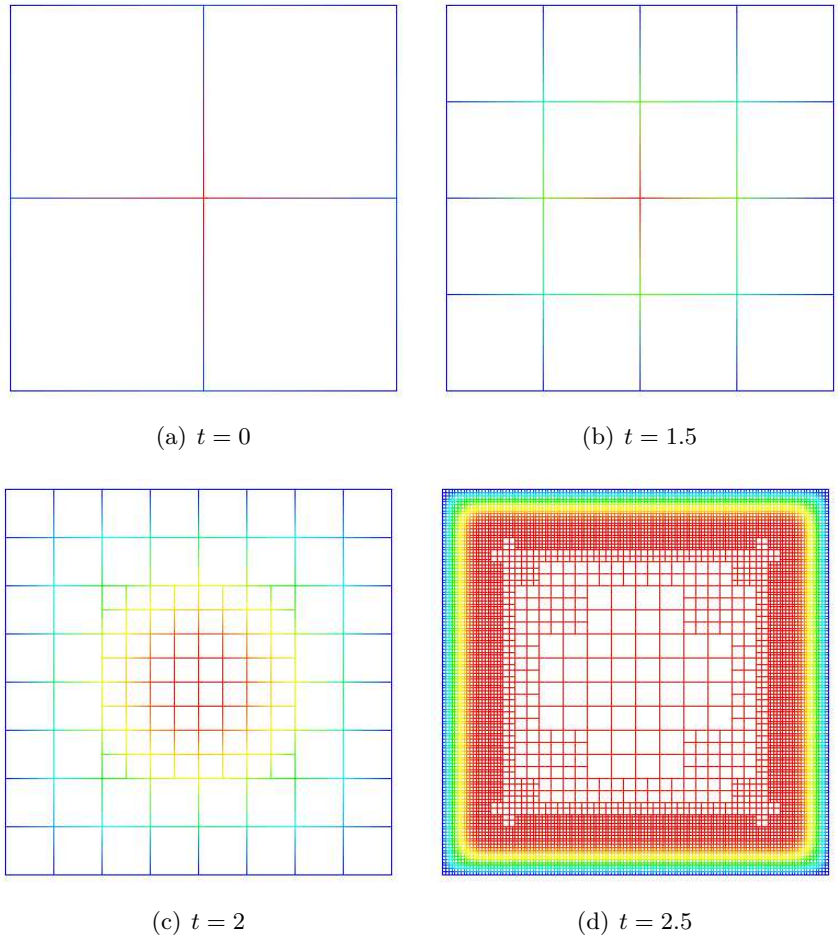


Figure 8.7: Discretization error estimate and projection error estimate with proceeding mesh refinement for Example 8.2. (x-axis: $\log(N)$, y-axis: $\log(|\eta_{\text{proj}}|)$ and $\log(|\eta_{\text{disc}}|)$)

The visualization of the meshes obtained by the procedure of local refinement at times $t = 0$, $t = 0.2$, $t = 0.4$, $t = 0.6$, $t = 0.8$ and $t = 1$ is given in Figure 8.9 at the end of the chapter on page 129. It illustrates the reason for the good performance of local refinement in comparison to global refinement: the singularities at the corners are resolved by locally refined meshes.

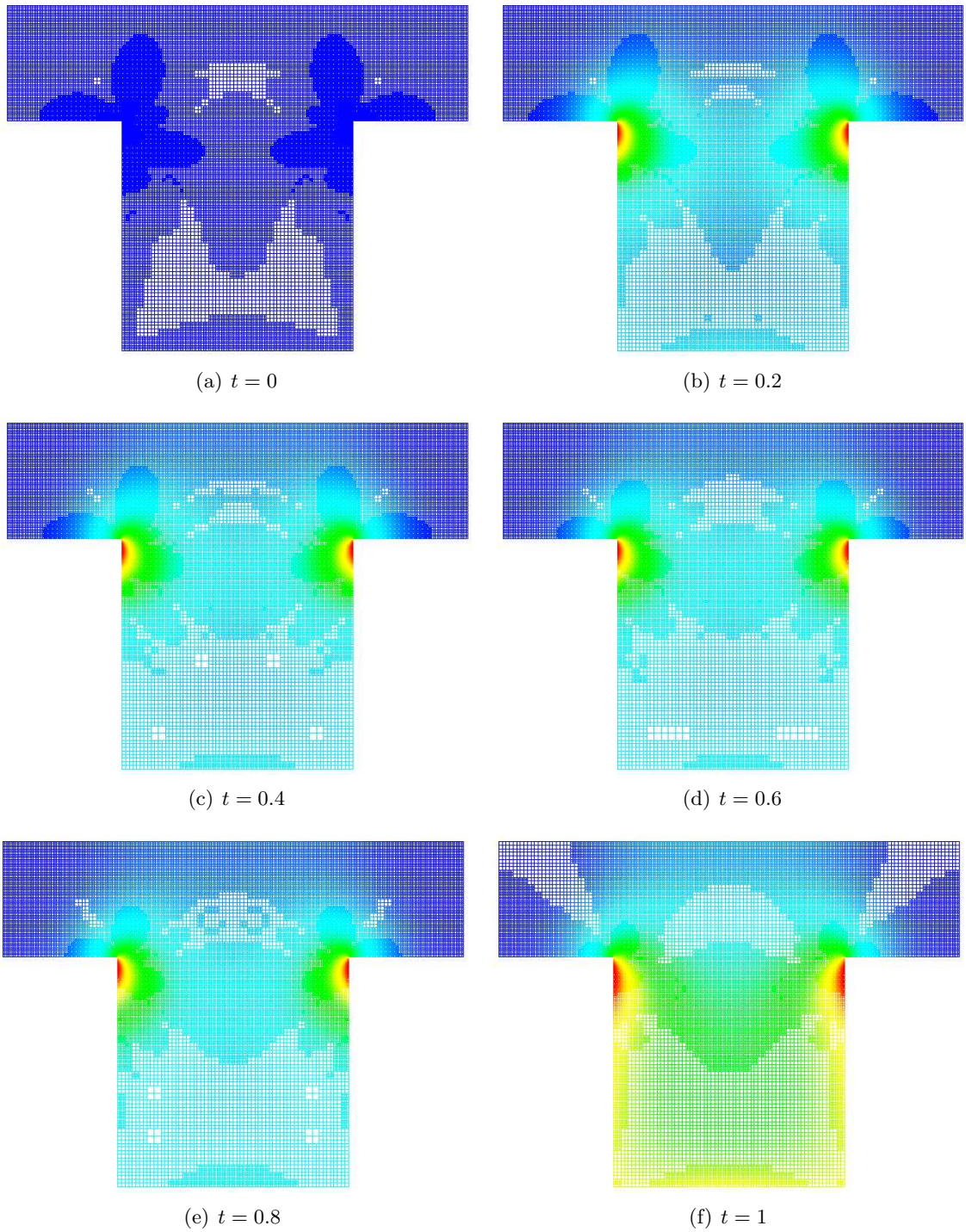
Now, we have the mesh adaptive multiple shooting method at hand and proceed in the next section with the consideration of the solid fuel ignition model in the context of multiple shooting.



(a) $t = 0$ (b) $t = 1.5$

(c) $t = 2$ (d) $t = 2.5$

Figure 8.8: Intervalwise constant meshes at $t = 0$, $t = 1.5$, $t = 2$, $t = 2.5$ for Example 8.1.



(a) $t = 0$ (b) $t = 0.2$

(c) $t = 0.4$ (d) $t = 0.6$

(e) $t = 0.8$ (f) $t = 1$

Figure 8.9: Intervalwise constant meshes at $t = 0$, $t = 0.2$, $t = 0.4$, $t = 0.6$, $t = 0.8$ and $t = 1$ for Example 8.2.

9 Application to the Solid Fuel Ignition Model

This chapter focuses on the application of mesh adaptive multiple shooting to the optimal control of the solid fuel ignition model. We introduce the solid fuel ignition model in Section 9.1 and briefly review the theoretical background in Section 9.2. Finally, in Section 9.3 we consider the optimal control of the model for different configurations and different types of cost functionals.

9.1 The Solid Fuel Ignition Model

The solid fuel ignition model originates from combustion theory for rapid exothermic chemical reactions. We consider the one-step irreversible reaction

$$\nu_F F + \nu_O O \rightarrow \nu_P P$$

which describes the oxidation of a fuel F with the oxidant O to the product P of this combustion process. In this connection, the stoichiometric constants ν_F, ν_O, and ν_P describe the absolute parts in which the reactants and products interact, and we additionally introduce the corresponding mass fractions of all species, y_F, y_O, and y_P. If both reactants F and O are available in correct proportions and the initial stock y_{F_0} and y_{O_0} is of the same order of magnitude then both species are consumed entirely during the process. Therefore, the reaction rate depends strongly on the mass fraction y_{F_0} *and* y_{O_0} which makes the mathematical description of the process unfortunately rather complicated. However, if we assume that the fuel F is available in excess, that is $y_{F_0} \gg y_{O_0}$, the mass fraction of F does not change significantly during the chemical process and can be assumed to be constant. With this simplifications at hand, the process reduces to a single species reaction determined by the development of y_O during the process of combustion. The derivation of the mathematical formulation of the reaction process is more elaborated and is performed by a step by step application of different conservation laws (that is the conservation of mass, the conservation of species, the conservation of momentum and the conservation of energy). We do not intend to give the explicit derivation here, but refer to the textbook of Bebernes and Eberly [2]. The so obtained equations describing the system are of a complex structure and incorporate not only the mass fractions, but also the density, the species velocity, the pressure, and the temperature. Since we are interested in the consideration of a solid fuel, which we assume to be a non deformable material of constant density, the equations simplify. First, the species velocity vanishes to zero, second the density can be assumed to be constant equal to 1. With

some additional minor simplifications we obtain the mathematical description of the solid fuel combustion with temperature T, time interval $I = (0, \infty)$, and fuel mass fraction y as

$$\begin{aligned} \partial_t T - \Delta T &= \varepsilon \delta y^m e^{\frac{T-1}{\varepsilon T}} && \text{in} \quad I \times \Omega, \\ \partial_t y - \beta \Delta y &= -\varepsilon \delta \Gamma y^m e^{\frac{T-1}{\varepsilon T}} && \text{in} \quad I \times \Omega \end{aligned}$$

with initial value and boundary condition given by

$$\begin{aligned} T(0,x) &= 1, \qquad y(0,x) = 1 \quad \text{in} \quad \Omega, \\ T(t,x) &= 1, \qquad \partial_n y(t,x) = 0 \quad \text{on} \quad I \times \partial\Omega. \end{aligned}$$

Here, $\beta \geq 0$, $\Gamma > 0$, and $\delta > 0$ is the so called Frank-Kamenetski parameter. For the fuels of interest, ε is small. Simplifying the model by the asymptotic first order approximation of temperature and mass fraction through

$$T = 1 + \varepsilon\theta \quad \text{and} \quad y = 1 - \varepsilon c$$

yields the reduced model

$$\begin{aligned} \partial_t \theta - \Delta \theta &= \delta (1 - \varepsilon c)^m e^{\frac{\theta}{1+\varepsilon\theta}} && \text{in} \quad I \times \Omega, \\ \partial_t c - \beta \Delta c &= -\delta \Gamma (1 - \varepsilon c)^m e^{\frac{\theta}{1+\varepsilon\theta}} && \text{in} \quad I \times \Omega \end{aligned}$$

with initial value and boundary condition

$$\begin{aligned} \theta(0,x) &= 0, \qquad c(0,x) = 0 \quad \text{in} \quad \Omega, \\ \theta(t,x) &= 0, \qquad \partial_n c(t,x) = 0 \quad \text{on} \quad I \times \partial\Omega. \end{aligned}$$

For $\varepsilon \ll 1$ the equations decouple and we are left with the consideration of the temperature θ only. This simplified system (9.3) is finally denoted as the *solid fuel ignition model* and describes under these assumptions the thermal reaction of a rigid material during the ignition period of the process.

$$\partial_t \theta - \Delta \theta = \delta e^{\theta} \quad \text{in} \quad I \times \Omega \tag{9.3a}$$

with initial value and boundary condition

$$\begin{aligned} \theta(0,x) &= 0 \qquad \text{in} \quad \Omega, \\ \theta(t,x) &= 0 \qquad \text{on} \quad I \times \partial\Omega. \end{aligned} \tag{9.3b}$$

The thermal single species reaction develops due to the energy which is set free during the reactive process and the equilibrating conduction of energy. Whenever the heat dissipation is sufficiently large compared to the released energy, an energetical equilibrium of the system can settle during the process of combustion. In this case the system does not heat up significantly because enough energy is conducted from the system due to cooling. However, if cooling does not provide an accordingly high heat dissipation, a localized temperature rise occurs and and the reaction is accelerated noticeably in this region. As a result, in this strongly localized high temperature region, the fuel is burned rapidly in an explosive reaction. Consequently the solid fuel combustion allows a distinction of two cases:

1. If the reaction settles into energetical equilibrium it is classified as *subcritical* or *fizzle* event.
2. If the reaction results in an explosive burst of power generation we speak of a *supercritical* or *explosive* event.

This classification is supported by theoretical results which provide information on the whereabouts of the appearance of the blowup. We finally introduce the *steady state solid fuel ignition model* which we need in the next section:

$$\begin{aligned} -\Delta\psi &= \delta e^{\psi} \qquad \text{in} \quad \Omega, \\ \psi(x) &= 0 \qquad\quad \text{on} \quad \partial\Omega. \end{aligned} \tag{9.4}$$

We give a short summary of the theoretical results mentioned above and review additionally the theory on existence and uniqueness of solutions from the literature in the following section.

9.2 Theoretical Background

We have explained so far that the solid fuel ignition model given in equation (9.3) has either a solution over the whole time or otherwise the solution becomes unbounded at a time $T < \infty$. This *blowup time* is mainly determined by a critical parameter δ_c, that is if $\delta < \delta_c$ the solution is bounded for $t \in (0, \infty)$, otherwise for $\delta > \delta_c$ the solution experiences a blowup at $t = T$. We start this section with the discussion of the existence and uniqueness of the solutions, before we proceed with some results providing an upper bound of the blowup time T in dependence on δ_c. The theoretical investigation is sophisticated and needs an extensive theoretical preparation. Therefore, we cite the essential results from [2] without proving them. Let δ_c denote the (reaction dependant) critical parameter. Then the following two theorems (Theorem 3.6 and 3.7 in [2]) hold:

Theorem 9.1. *For $\delta < \delta_c$, problem* (9.3) *has a unique solution $\theta(t,x) \in C^{1,2}(I \times \Omega)$ with $0 \le \theta(t,x) \le \varphi(x)$, where $\varphi(x)$ is the minimal solution of the steady state problem* (9.4)*.*

Remark 9.1. With the assumptions of Theorem 9.1 we obtain additional information on the convergence behavior of the solution in time which is

$$\lim_{t\to\infty} \theta(t,x) = \varphi(x).$$

Theorem 9.2. *For each $\delta > \delta_c$, there is a $T \in [1/\delta, \infty)$ such that* (9.3) *has a unique solution $\theta(t,x) \in C^{1,2}((0,T) \times \Omega)$. Moreover,*

$$\lim_{t \nearrow T} \max \left\{ \theta(t,x) \;\middle|\; x \in \bar{\Omega} \right\} = \infty.$$

In other words, the solution $\theta(t,x)$ of (9.3) is globally existent and converges to the minimal solution of the steady state problem (9.4) if the parameter δ stays below the critical bound δ_c. Otherwise, the solution becomes unbounded in the L^∞-sense as $t \nearrow T$.

Further research allows not only to decide whether a system exhibits a blowup, but also to determine for given δ_c the maximum time interval for which the solution exists. We cite without proof from [2] the following two theorems and a corollary answering this question. The basic idea is to find an ODE initial value problem whose solution exhibits a blowup in time and is a lower bound for the solution to the solid fuel ignition model:

Theorem 9.3. *(Theorem 3.9 from [2]) Let $u(t)$ be the solution of the initial value problem*

$$u' = \delta e^u - \lambda_1 u, \quad t \in (0,T) \quad \text{and} \quad u(0) = 0$$

where λ_1 is the first eigenvalue of $-\Delta\varphi = \lambda\varphi$, $x \in \Omega$ and $u(0) = 0$, $x \in \partial\Omega$. Let $\theta(t,x)$ be the solution of (9.3) on $[0,T) \times \bar{\Omega}$, then

$$u(t) \leq \sup \left\{ \theta(t,x) \;\middle|\; x \in \bar{\Omega} \right\}$$

for $t \in [0,T)$.

It is easy to verify that the solution u only exists on the time interval $[0,T_0)$ where $u(t) \to \infty$ for $t \nearrow T_0$. The maximum time T_0 is determined by the identity

$$T_0 = \int_0^\infty \frac{dz}{\delta e^z - \lambda_1 z}.$$

Consequently, $T_0 < \infty$ if $\delta > \delta^* := \lambda_1/e$, and in this case the blowup occurs in finite time as stated in the corollary below.

Corollary 9.4. *(Corollary 3.10 from [2]) If $\delta > \delta^* = \lambda_1/e$, then $T_0 < \infty$ and*

$$\lim_{t \nearrow T_0} \sup \left\{ \theta(t,x) \;\middle|\; x \in \bar{\Omega} \right\} = \infty$$

where $T < T_0$. That is, the blowup occurs in finite time.

We substantiate this result by means of a numerical example. Considering $\Omega = (-1,1)\times(-1,1)$ we obtain for the smallest eigenvalue of the Laplacian $\lambda_1 = \frac{\pi^2}{2}$ and for the parameter $\delta^* = 1.815$. Furthermore, the critical parameter is given by $\delta_c = 1.734$. We can now investigate for different parameters $\delta > \delta^*$ the quality of the upper bound T_0 for the blowup time T. Figure 9.1 shows the behavior of predicted and numerically computed blowup time for different reaction parameters δ. The corollary states that for $\delta > \delta^* > \delta_c$ the solution exhibits a blowup in finite time and further more gives an upper bound for the blowup time. For the numerical example considered in the next section, we ensure that $\delta > \delta^*$ and choose the time interval $I = (0,T)$ such that $(0,T_0) \subset I$. For the sake of completeness, we finally present results for the blowup time for those remaining parameters $\delta \in (\delta_c, \delta^*)$. Bebernes and Eberly derive for this case the following theorem:

Theorem 9.5. *(Theorem 3.13 from [2]) If δ_c is in the spectrum of (9.4) and if $\delta > \delta_c$, then the unique solution $\theta(t,x)$ of (9.3) blows up in finite time T where*

$$T < \sqrt{\frac{2\pi^2}{\delta_c(\delta - \delta_c)}}.$$

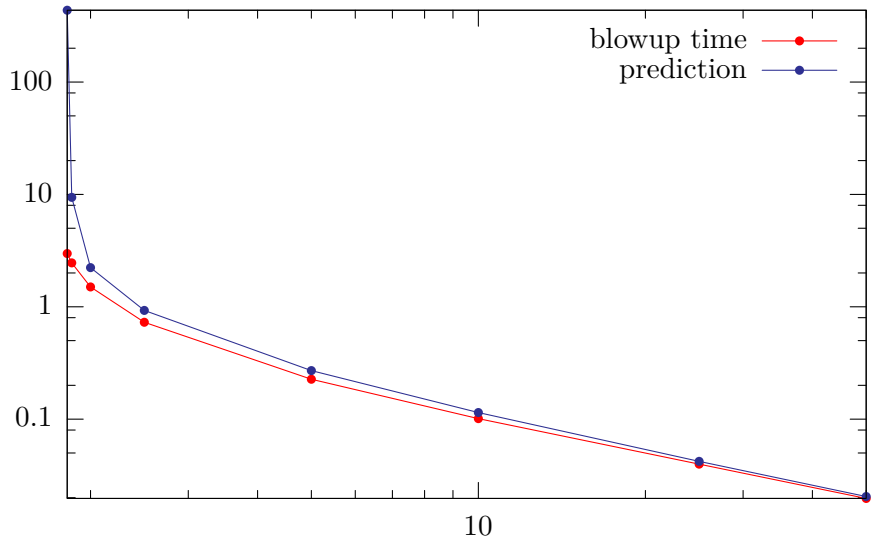


Figure 9.1: Predicted and calculated blowup time for different parameters δ with $\delta^* = 1.815$ and $\delta_c = 1.734$. (x-axis: $\log(\delta)$, y-axis: $\log(t)$)

With the theoretical results at hand, we are able to construct an example whose solution blows up in finite time. Against this background, our applications tend to control this explosive system by controlling the heat dissipation. That is, we choose the control as a source term on the right hand side of equation (9.3a).

9.3 Optimal Control of the Solid Fuel Ignition Model

Optimal control of the solid fuel ignition model is of practical relevance in chemical engineering. We consider here the 2-dimensional case on a squared domain with distributed control for different cost functionals.

Example 9.1. (Distributed control of the 2-dimensional solid fuel ignition model) Let $\Omega = (-1,1) \times (-1,1) \subset \mathbb{R}^2$ be given, and chose $\delta = 2.5$ for the fuel ignition model. By means of Theorem 9.3 and Corollary 9.4 we obtain that in this configuration the solution exhibits a blowup at $T < T_0 = 0.929$. We define the time dependent discontinuous function

$$\bar{\theta}(t,x,y) := \begin{cases} t \cdot \cos\left(\frac{\pi}{2}x\right) \cdot \cos\left(\frac{\pi}{2}y\right) & \text{for} \quad t < 1, \\ t \cdot \cos\left(3\frac{\pi}{2}x\right) \cdot \cos\left(3\frac{\pi}{2}y\right) & \text{for} \quad t \geq 1, \end{cases}$$

the time dependent continuous function

$$\hat{\theta}(t,x,y) := t \cdot \cos\left(3\frac{\pi}{2}x\right) \cdot \cos\left(3\frac{\pi}{2}y\right),$$

and the time independent function

$$\tilde{\theta}(x, y) := 0.5.$$

In the following, we consider three different cost functionals:

1. A time distributed cost functional of the structure

$$J(q, \theta) := \frac{1}{2} \int_I \|\theta(t) - \bar{\theta}(t)\|^2 dt + \frac{10^{-3}}{2} \|q(t)\|_Q^2. \tag{9.5}$$

2. A terminal time cost functional

$$J(q, \theta) := \frac{1}{2} \|\theta(T) - \tilde{\theta}\|^2 + \frac{10^{-3}}{2} \|q(t)\|_Q^2. \tag{9.6}$$

3. Another distributed cost functional given by

$$J(q, \theta) := \frac{1}{2} \int_I \|\theta(t) - \bar{\theta}(t)\|^2 dt + \frac{1}{2} \int_I \|e^{\theta(t)} - e^{\bar{\theta}(t)}\|^2 dt + \frac{10^{-3}}{2} \|q(t)\|_Q^2. \tag{9.7}$$

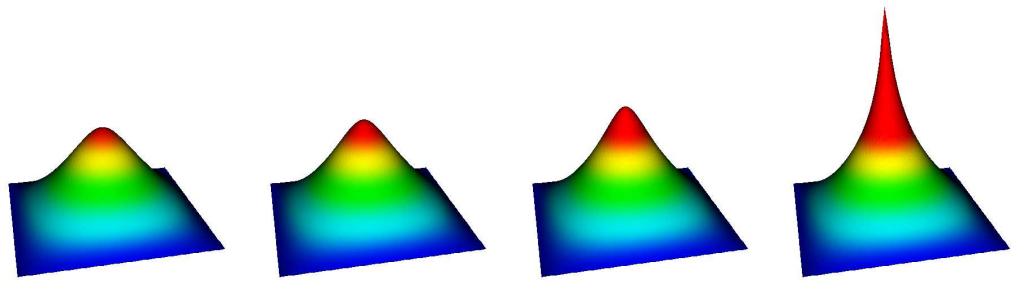
We obtain the optimal control problem of interest as

$$\min_{q \in Q} J(q, \theta)$$

such that

$$\begin{aligned} \partial_t \theta - \Delta\theta - \delta e^\theta &= q, && \text{in} \quad I \times \Omega, \\ \theta(0, x) &= 0 && \text{in} \quad \Omega, \\ \theta(t, x) &= 0 && \text{on} \quad I \times \partial\Omega. \end{aligned}$$

First of all, let us consider the forward simulation of the problem, that is the solution of problem (9.3) on the given domain Ω for $q = 0$. The calculation breaks down at $t = 0.715$, and the development of the solution towards this point is shown in Figure 9.2. The solution

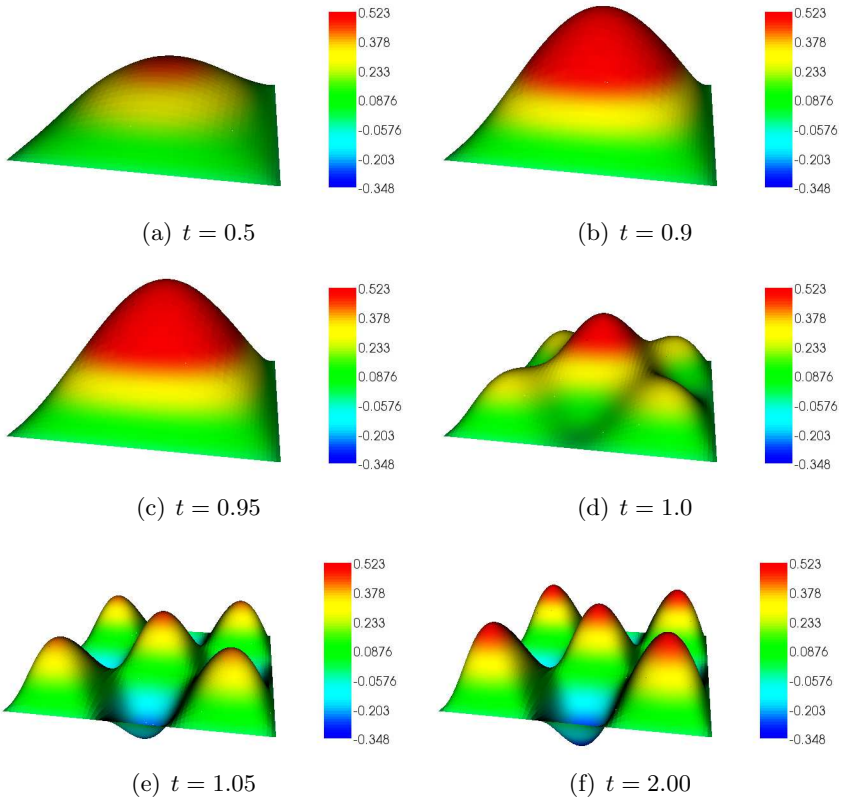


(a) $t = 0.706$ (b) $t = 0.709$ (c) $t = 0.712$ (d) $t = 0.715$

Figure 9.2: Solution for the forward simulation of Example 9.1.

blows up at approximately $t = 0.715$. A forward simulation with $q = 0$ and several other

tested initial controls on the whole time interval is not possible, and therefore common solution techniques which require the solution of the problem on the whole time interval I for the calculation of the control update can not be applied. As outlined in the introduction, multiple shooting overcomes this difficulty and allows the calculation of piecewise solutions. For the cost functional (9.5) we apply multiple shooting on the time interval $I = (0, 2)$ with 40 multiple shooting intervals each consisting of 5 time steps of length 0.01. The solution at different timepoints is given in Figure 9.3. All meshes are refined locally, such that the solutions on different intervals are computed on differently fine discretizations. Additionally, we show the control at the same timepoints and thereby the corresponding meshes in Figure 9.4.



(a) $t = 0.5$ (b) $t = 0.9$

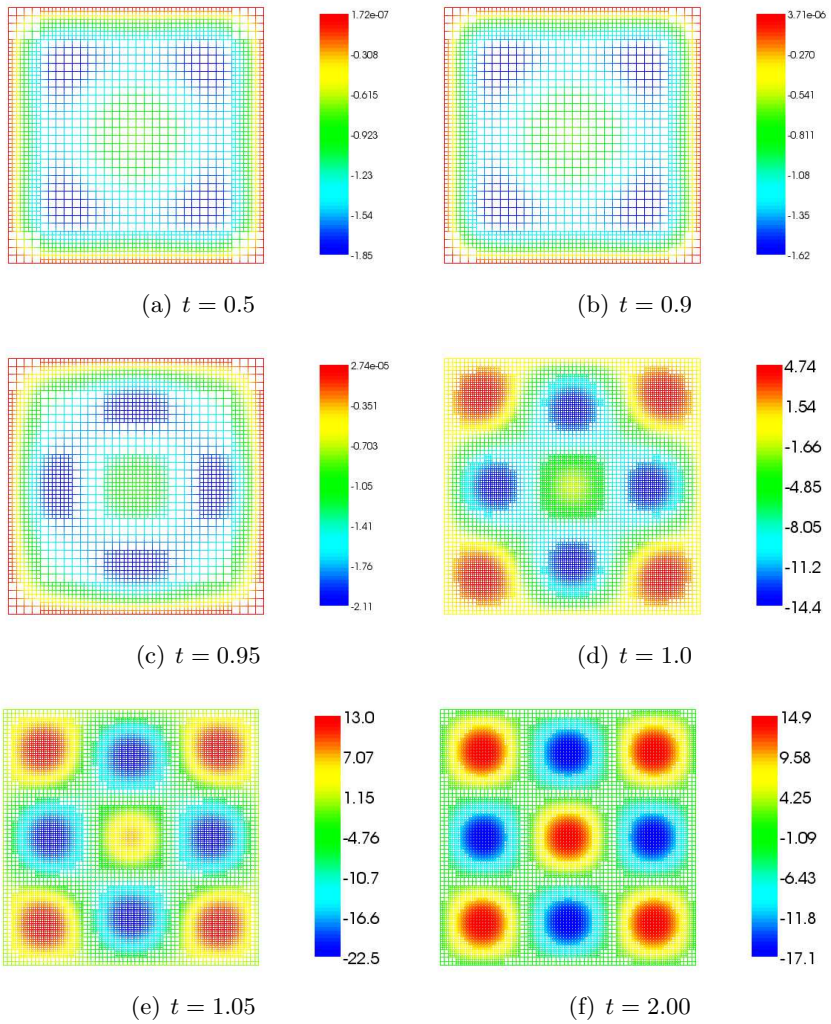
(c) $t = 0.95$ (d) $t = 1.0$

(e) $t = 1.05$ (f) $t = 2.00$

Figure 9.3: Solution for Example 9.1 with cost functional (9.5).

Extrapolating the exact minimal value of the functional as 0.775194, we can compare the efficiency of local and global refinement. The choice of local refinement is for this example of less importance than in those examples considered in the previous chapter. In fact, no singularities or traveling fronts have to be resolved such that local refinement yields only a slight improvement for this example as illustrated in Figure 9.5.

This is different for the case of the terminal time cost functional (9.6). We consider the time interval $I = (0, 1)$ for which a multiple shooting time domain decomposition of 10 subintervals each consisting of 10 time steps of length 0.01 is chosen. As expected, due to the Dirichlet boundary data, local refinement takes place mainly on the last time interval.



(a) $t = 0.5$ (b) $t = 0.9$

(c) $t = 0.95$ (d) $t = 1.0$

(e) $t = 1.05$ (f) $t = 2.00$

Figure 9.4: Control and meshes for Example 9.1 with cost functional (9.5).

It results in a thin boundary layer of refined cells. The exact functional value is obtained from extrapolation as 0.10116413 for which we compare the performance for local and global refinement in Figure 9.6. The solution at different time points for this optimization problem is shown in Figure 9.7, and as before the control and mesh are given afterwards in Figure 9.8.

Finally, we consider the problem for the third cost functional (9.7). In this case, we decompose $I = (0, 1)$ into 10 multiple shooting intervals and apply as before the implicit Euler time stepping scheme with step size 0.01. We obtain an approximation of the exact functional by extrapolation as 0.09053600. As in the first configuration, local refinement yields only slight improvements of the efficiency. This result is shown in Figure 9.9. The obtained solution and control at different time points are shown at the end of this chapter in Figures 9.10 and 9.11 in which the structure of the refined meshes is outlined.

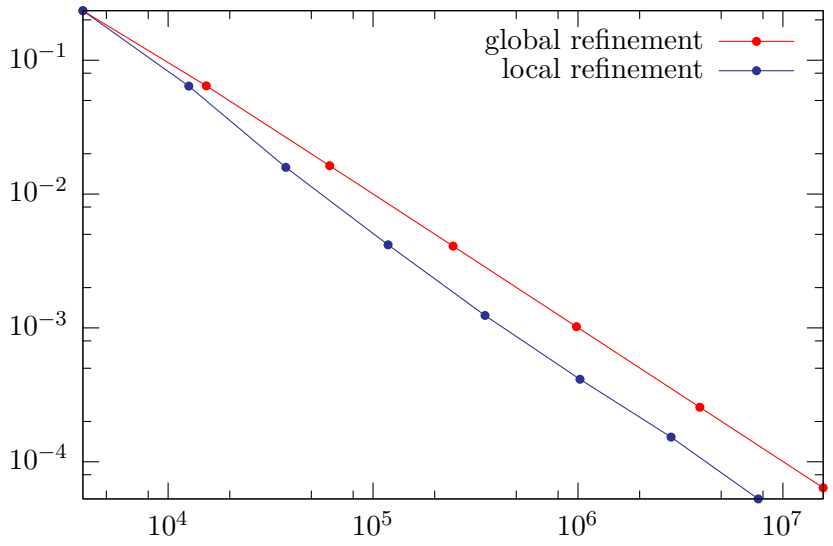


Figure 9.5: Functional error for differently fine meshes, gained by local and global mesh refinement for Example 9.1 with cost functional (9.5). (x-axis: $\log(N)$, y-axis: $\log(|e_h|)$)

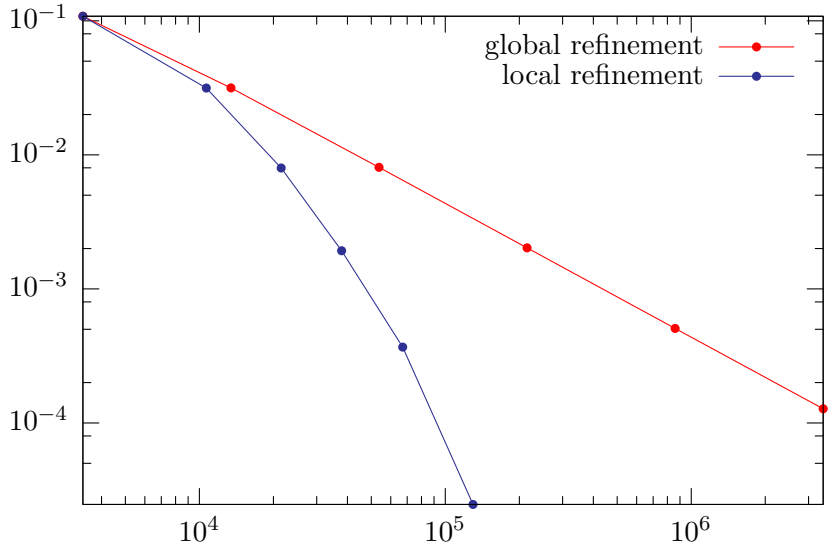
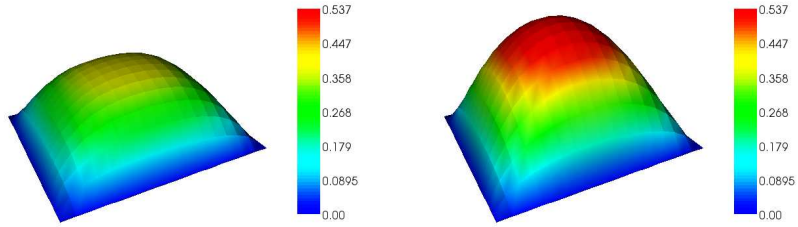
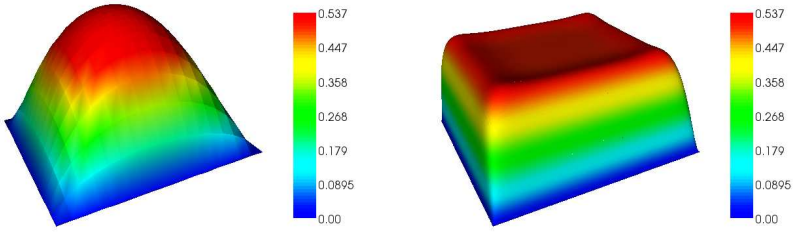


Figure 9.6: Functional error for differently fine meshes, gained by local and global mesh refinement for Example 9.1 with cost functional (9.6). (x-axis: $\log(N)$, y-axis: $\log(|e_h|)$)

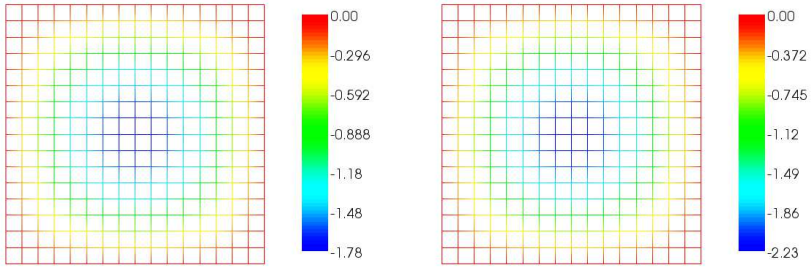


(a) $t = 0.25$ (b) $t = 0.5$

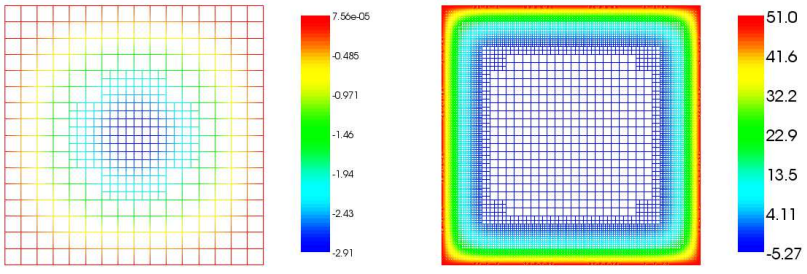


(c) $t = 0.75$ (d) $t = 1.00$

Figure 9.7: Solution for Example 9.1 with cost functional (9.6).



(a) $t = 0.25$ (b) $t = 0.5$



(c) $t = 0.75$ (d) $t = 1.00$

Figure 9.8: Control and meshes for Example 9.1 with cost functional (9.6).

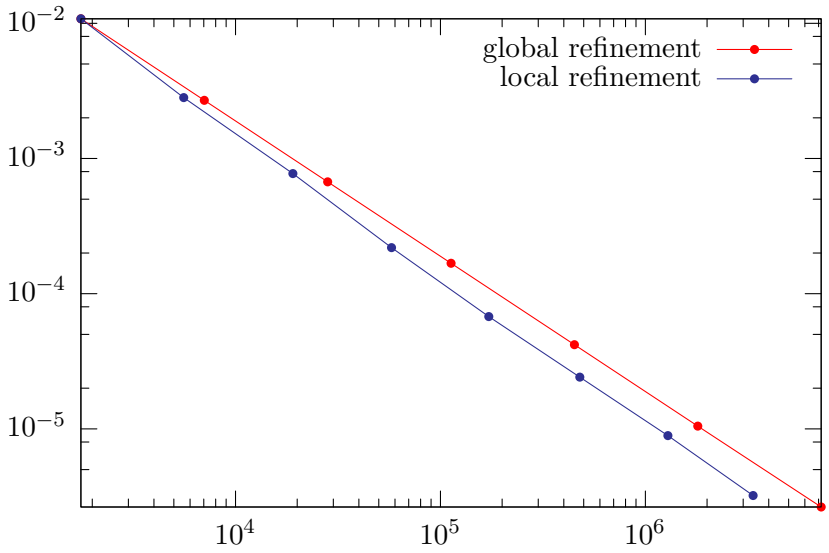
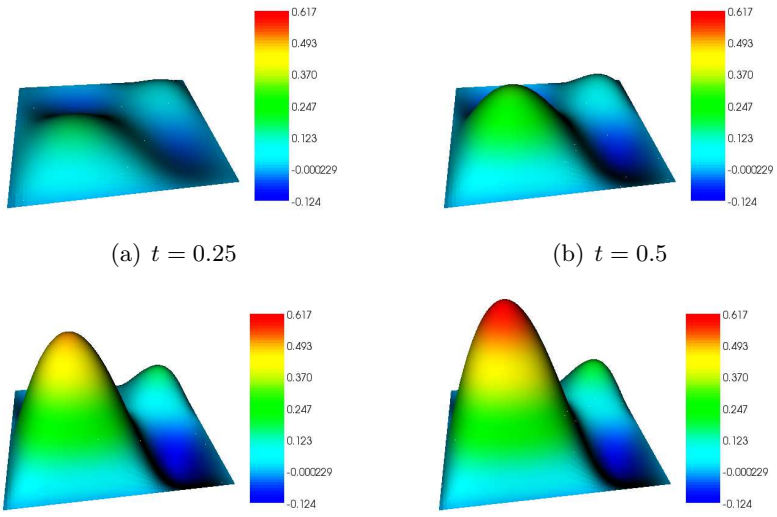


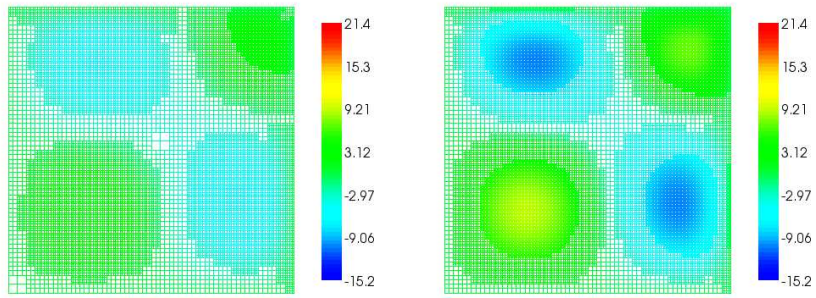
Figure 9.9: Functional error for differently fine meshes, gained by local and global mesh refinement for Example 9.1 with cost functional (9.7). (x-axis: $\log(N)$, y-axis: $\log(|e_h|)$)



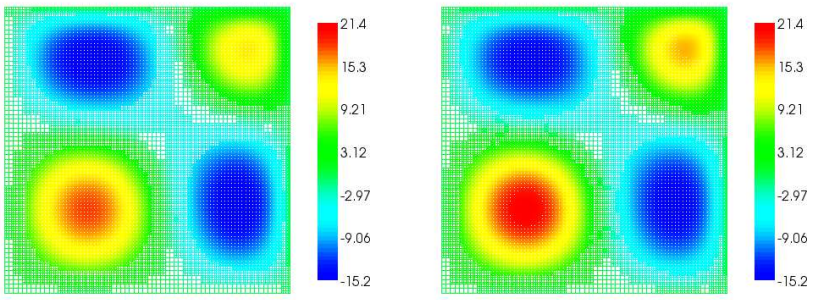
(a) $t = 0.25$ (b) $t = 0.5$

(c) $t = 0.75$ (d) $t = 1.00$

Figure 9.10: Solution for Example 9.1 with cost functional (9.7).



(a) $t = 0.25$ (b) $t = 0.5$



(c) $t = 0.75$ (d) $t = 1.00$

Figure 9.11: Control and meshes for Example 9.1 with cost functional (9.7).

10 Conclusion and Outlook

In this thesis we developed multiple shooting approaches for optimization problems constrained by partial differential equations. Additionally, we investigated the application of a posteriori error estimation and spatial mesh adaptation in the context of multiple shooting.

Starting from the multiple shooting approach for ordinary differential equations, we extended the direct and indirect multiple shooting approach to problems constrained by partial differential equations. In this context, we also outlined the relation between direct and indirect multiple shooting. Consequently, the development of the approaches also involved discussing the general differences between the ODE and the PDE constrained case. We concluded that the PDE case has several limitations with respect to the efficiency. While the ODE case exploits efficient derivative generation techniques together with fast explicit assembling of the system matrices, the PDE case does not allow this procedure. The reason for this restriction is the extremely high dimension of the system matrices due to fine spatial discretizations. In this context, possible extensions might lead into the direction of reduction techniques. On the one hand, exploiting a priori information on the problem might yield a reduction of the control space and, on the other hand, a coarser problem suited discretization of the control might reduce the dimension of the control space even further. These ideas might therefore allow to accelerate the condensed approach further.

We discussed that optimization problems with highly unstable PDE constraints do not permit the solution over the whole time interval. These problems cannot be solved by standard solution techniques. Multiple shooting as a time domain decomposition technique is a suitable solution approach for the solution of such problems as we saw for the explosive system given by the solid fuel ignition model.

We considered different solution techniques applied to direct and indirect multiple shooting. We started from Newton's method for the solution of the system of matching conditions, proceeded with a preconditioned GMRES method for the linearized system, and finally concluded with the investigation of different solvers for the intervalwise problems. These problems comprised linear and nonlinear initial value problems as well as linear and nonlinear boundary value problems. We discovered, that most of the computation time is spent on the generation of the directional derivatives needed for the iterative solution of the linearized system. Numerical tests indicated that preconditioning is inevitable but increases the numerical effort noticeably. Condensing techniques were introduced as a possible alternative to save computational effort and time. Promising topics for further research seem to be preconditioning techniques for the condensed approach and additional preconditioners, preferably parallelizable, for the direct approach.

Furthermore, we studied the combination of multiple shooting with a posteriori error estimation: first the application of the standard approach, and second the development of an

error estimator suited for the needs of multiple shooting with different adjacent meshes on the multiple shooting nodes. Further future developments in this area of research might lead into the direction of time adaptive strategies. On the one hand, a posteriori error estimation can be applied for the temporal error, and mesh adaptation in time follows straightforward. On the other hand, an appropriate splitting of the multiple shooting intervals would not only allow the reduction of the temporal error, but also the reduction of the spatial error in the case of intervalwise constant meshes. The idea of finding a diagonal sequence of spatial meshes within the outer Newton method should also be investigated. This procedure performs mesh refinement whenever the error due to the Newton residual becomes smaller than the discretization error. Thus, the number of Newton steps needed during the solution procedure might be reduced.

Finally, after having understood the general features of multiple shooting with mesh adaptation for PDE constrained optimization problems, the application of the developed multiple shooting approach to sophisticated application problems is advisable.

Acknowledgments

This work was supported by the German Research Foundation (DFG) through the *International Graduiertenkolleg* "Complex processes: Modeling, Simulation and Optimization". I am very grateful that the German Research Foundation offered me the opportunity to do my research in an interdisciplinary graduate school which offered not only generous funding and provided an insight into different areas of scientific research, but also allowed me to participate in various workshops abroad and to stay in the United States for a two month research visit.

I want to express my gratitude to my supervisors Prof. Dr. Dr. h.c. Hans Georg Bock, *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR)*, University of Heidelberg, and Prof. Dr. Rolf Rannacher, *Institut für Angewandte Mathematik*, University of Heidelberg, for providing this interesting subject and for their support and many fruitful discussions over the past three years.

During this time, many other people have contributed to this work. First and foremost, I want to thank Prof. Dr. Guido Kanschat, *Department of Mathematics*, Texas A&M University, for introducing me into the finite element software package deal.II, for his continuous support and innovative ideas, for the many interesting and encouraging conversations and last but not least for the warm welcome during my stay at Texas A&M University. Furthermore, I would like to express my gratitude to Dr. Johannes Schlöder, *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR)*, University of Heidelberg, for his support during the early stage of this work. I am indebted to Prof. Dr. Matthias Heinkenschloss, *Department of Computational and Applied Mathematics*, Rice University, who kindly offered me to stay with his group for two weeks and provided me not only an interesting insight into his previous work on the subject, but also took the time to critically discuss and improve my ideas on the topic.

I also want to thank my colleague Dr. Dominik Meidner for his advise in many computational matters and my office mate Bärbel Janssen for the warm and pleasant atmosphere and the interesting discussions on the deal.ii software package.

Finally, I want to thank my friends and my family: my sister, Dr. Kerstin Hesse, *Department of Mathematics*, University of Sussex, for proofreading and my parents for the wonderful encouraging support.

Bibliography

[1] J. Albersmeyer. *Effiziente Ableitungserzeugung in einem adaptiven BDF-Verfahren.* Diploma thesis, University of Heidelberg, 2005.

[2] J. Bebernes and D. Eberly. *Mathematical Problems from Combustion Theory.* Springer, Berlin, Heidelberg, New York, 1989.

[3] R. Becker. *Adaptive Finite Elements for Optimal Control Problems.* Habilitation thesis, University of Heidelberg, 2001.

[4] R. Becker and H. Kapp. Optimization in PDE models with adaptive finite element discretization. In H. G. Bock, F. Brezzi, R. Glowinski, G. Kanschat, Y. A. Kuznetsov, J. Périaux, and R. Rannacher, editors, *Numerical Mathematics and Advanced Applications*, pages 147–155. World Scientific, London, 1998.

[5] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimal control of partial differential equations: basic concepts. *SIAM J. Control Optim.*, 31:113–132, 2000.

[6] R. Becker, D. Meidner, and B. Vexler. Efficient Numerical Solution of Parabolic Optimization Problems by Finite Element Methods. *Optimization Methods and Software*, 22(5):813–833, 2007.

[7] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In *Acta Numerica*, 10, pages 1–102. Cambridge University Press, 2001.

[8] H. G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K. H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, 18 of *Springer Series in Chemical Physics*, pages 102–125. Springer, Heidelberg, 1981.

[9] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, 183 of *Bonner Mathematische Schriften.* University of Bonn, 1987.

[10] H. G. Bock, M. Diehl, D. B. Leineweber, and J. P. Schlöder. Efficient direct multiple shooting in nonlinear model predictive control. In F. Keil, W. Mackens, H. Voss, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, 2, pages 218–227, Berlin, 1999. Springer.

[11] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, pages 243–247. Pergamon Press, 1984.

[12] D. Braess. *Finite Elemente, Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, Berlin, Heidelberg, New York, 2nd edition, 1997.

[13] R. Bulirsch. Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung. Technical report, Carl-Crantz-Gesellschaft, 1971.

[14] A. Comas. *Time-Domain Decomposition Preconditioners for the Solution of Discretized Parabolic Optimal Control Problems*. PhD thesis, Rice University, 2005.

[15] B. Dacorogna. *Direct Methods in the Calculus of Variations*, 78 of *Appl. Math. Sci.* Springer, 1989.

[16] R. Dautray and J. L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology: Evolution Problems I*, 5. Springer, 1992.

[17] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Press Syndicate of the University of Cambridge, 1996.

[18] A. V. Fursikov. Optimal Control of Distributed Systems: Theory and Applications. *Transl. Math. Monogr.*, 187, 1999.

[19] A. Hanf. *Konstruktion eines Hochtemperaturströmungsreaktors und Untersuchung der Reaktion* $O(^1D) + H_2 \rightarrow OH + H$ *bei hohen Temperaturen*. PhD thesis, University of Heidelberg, 2005.

[20] M. Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *J. comput. appl. math.*, 173:169–198, 2005.

[21] H. K. Hesse and G. Kanschat. Mesh adaptive multiple shooting for partial differential equations. Part I: Linear quadratic optimal control problems. *submitted*.

[22] M. Hinze. A variational discretization concept in control constrained optimization: The linear quadratic case. *Comput. Optim. Appl.*, 30(1):45–61, 2005.

[23] J. F. Holt. Numerical solution of nonlinear two-point boundary problems by finite difference methods. *Communications of the ACM*, 7:366–273, 1964.

[24] K. Ito and K. Kunisch. Optimal control of the solid fuel ignition model with H^1-cost. *SIAM J. Control Optim.*, 40(5):1455–1472, 2002.

[25] A. Kauffmann. *Optimal Control of the Solid Fuel Ignition Model*. PhD thesis, Technical University of Berlin, 1998.

[26] H. B. Keller. *Numerical Solution of Two Point Boundary Value Problems*, 24, pages 1–19. Society for Industrial and Applied Mathematics, 1976.

[27] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects. *Comp. Chem. Eng.*, 27:157–166, 2003.

[28] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*, 170 of *Grundlehren Math. Wiss.* Springer, 1971.

[29] D. Meidner. *Adaptive Space-Time Finite Element Methods for Solving Optimization Problems Governed by Parabolic Systems.* PhD thesis, University of Heidelberg, 2007.

[30] D. Meidner and B. Vexler. Adaptive space-time finite element methods for parabolic optimization problems. *SIAM J. Control Optim.*, 46(1):116–142, 2007.

[31] D. D. Morrison, J. D. Riley, and J. F. Zancanaro. Multiple shooting method for two-point boundary value problems. *Communications of the ACM*, 5:613–614, 1962.

[32] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, 1999.

[33] K. J. Plitt. *Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen.* Diploma thesis, University of Bonn, 1981.

[34] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Pub. Co., Boston, 1996.

[35] A. Schäfer, P. Kuehl, M. Diehl, J. P. Schlöder, and H. G. Bock. Fast reduced multiple shooting methods for nonlinear model predictive control. *Chemical Engineering and Processing*, 46(11):1200–1214, 2007.

[36] A. A. S. Schäfer. *Efficient Reduced Newton-type Methods for Solution of Large-Scale Structured Optimization Problems with Application to Biological and Chemical Processes.* PhD thesis, University of Heidelberg, 2005.

[37] R. Serban, S. Li, and L. Petzold. Adaptive algorithms for optimal control of time-dependent partial differential-algebraic equation systems. *Internat. J. Numer. Methods Engrg.*, 57(10):1457–1469, 2003.

[38] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20(3):626–637, 1983.

[39] J. Stoer and R. Bulirsch. *Einführung in die Numerische Mathematik II.* Springer Verlag, 1973.

[40] V. Thomée. *Galerkin Finite Element Methods for Parabolic Equations*, 25 of *Springer Series in Computational Mathematics.* Springer, 1997.

[41] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen. Theorie, Verfahren und Anwendungen.* Vieweg & Sohn, 2005.

[42] S. Ulbrich. Generalized SQP-methods with "parareal" time-domain decomposition for time-dependent PDE-constrained optimization. Technical report, Fachbereich Mathematik TU Darmstadt, 2005.

[43] J. Wloka. *Partielle Differentialgleichungen. Sobolevräume und Randwertaufgaben.* Teubner, 1982.