Inaugural-Dissertation

zur Erlangung der Doktorwürde der

Naturwissenschaftlich - Mathematischen
Gesamtfakultät

der

Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Diplom-Mathematiker

Michael Ernst Geiger

aus Aschaffenburg

Tag der mündlichen Prüfung: _____

Adaptive Multiple Shooting for Boundary Value Problems and Constrained Parabolic Optimization Problems

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

Abstract

Subject of this thesis is the development of adaptive techniques for multiple shooting methods. The focus is on the application to optimal control problems governed by parabolic partial differential equations. In order to retain as much freedom as possible in the later choice of discretization schemes, the details of both direct and indirect multiple shooting variants are worked out on an abstract function space level. Therefore, shooting techniques do not constitute a way of discretizing a problem. A thorough examination of the connections between the approaches provides an overview of different shooting formulations and enables their comparison for both linear and nonlinear problems.

We extend current research by considering additional constraints on the control variable in the multiple shooting context. An optimization problem is developed which includes so-called box constraints in the multiple shooting context. Several modern algorithms treating control constraints are adapted to the requirements of shooting methods. The modified algorithms permit an extended comparison of the different shooting approaches.

The efficiency of numerical methods can often be increased by developing grid adaptation techniques. While adaptive discretization schemes can be readily transferred to the multiple shooting context, questions of conditioning and stability make it difficult to develop adaptive features for shooting point distribution in multiple shooting processes. We concentrate on the design and comparison of two different approaches to shooting grid adaptation in the framework of ordinary differential equations. A residual-based adaptive algorithm is transferred to parabolic optimization problems with control constraints.

The presented concepts and methods are verified by means of several examples, whereby theoretical results are numerically confirmed. We choose the test problems so that the simple shooting method becomes unstable and therefore a genuine multiple shooting technique is required.

Zusammenfassung

Gegenstand dieser Arbeit ist die Entwicklung adaptiver Techniken für Mehrfachschießmethoden. Im Fokus liegt hierbei die Anwendung auf Optimalsteuerungsprobleme, welche durch parabolische partielle Differentialgleichungen beschränkt sind. Um möglichst viel Freiheit bei der späteren Wahl von Diskretisierungsschemata zu bewahren, werden die Details von direkten wie indirekten Verfahrensvarianten im abstrakten Funktionenraum ausgearbeitet. Schießverfahren stellen daher keine Diskretisierungsmethode dar. Eine eingehende Untersuchung der Zusammenhänge zwischen den Ansätzen liefert eine Übersicht der verschiedenen Verfahrensformulierungen und ermöglicht ihren Vergleich im Rahmen von linearen wie nichtlinearen Problemstellungen.

Wir erweitern den aktuellen Forschungsstand, indem wir zusätzliche Beschränkungen an die Steuervariable im Kontext von Mehrfachschießverfahren betrachten. Unter Einbezug sogenannter Box-Schranken wird zunächst ein Optimierungsproblem im Rahmen von Mehrfachschießmethoden entwickelt. Mehrere moderne Algorithmen zur Behandlung von Steuerungsbeschränkungen werden an die Bedürfnisse der Schießverfahren angepasst. Für die modifizierten Verfahren wird dann ein erweiterter Vergleich der unterschiedlichen Schießverfahren vorgenommen.

Vielfach lässt sich die Effizienz numerischer Verfahren durch Entwicklung von Techniken zur Gitteradaption steigern. Während sich adaptive Diskretisierungsschemata ohne Weiteres in den Kontext

von Schießverfahren einbetten lassen, wird die adaptive Steuerung der Schießpunkte bei Mehrfachschießprozessen durch Konditionierungs- und Stabilitätsfragen erschwert. Wir konzentrieren uns auf die Entwicklung und den Vergleich zweier verschiedener Ansätze zur Schießgitteradaption im Kontext gewöhnlicher Differentialgleichungen. Ein residuenbasierter adaptiver Algorithmus wird auf Optimierungsprobleme mit parabolischen Nebenbedingungen und beschränkten Steuervariablen übertragen.

Die vorgestellten Konzepte und Methoden werden anhand mehrerer Testbeispiele überprüft, und theoretische Resultate werden so numerisch bestätigt. Dabei werden insbesondere Probleme gewählt, für die das Einfachschießverfahren instabil ist und die daher ein echtes Mehrfachschießverfahren erfordern.

Contents

1 Introduction

Within the general framework of current research on optimization with partial differential equations (PDE), this thesis focusses on the application of multiple shooting methods to parabolic optimal control problems (OCP). Although shooting methods are a standard solution routine for boundary value problems (BVP) and optimization problems governed by ordinary differential equations (ODE), their employment in the PDE context is still in the early stages.
A suggestive way of applying multiple shooting to nonstationary PDE is to discretize the PDE by a method of lines (MOL) approach. The semidiscretization in space leads to a large system of ODE which is then solved by standard routines. It is straightforward to include multiple shooting methods into this extended ODE framework. However, the development of appropriate solvers for the discretized problems is then a delicate matter (see, e. g., the work of Potschka [94]). As an alternative, we propose to develop a multiple shooting reformulation of a given nonstationary PDE problem on the abstract function space level. This leaves more freedom for the later choice of discretization schemes, particularly for using adaptive discretization concepts in the framework of Rothe's method. Adaptive features of this kind are crucial for complex (e. g., spatially three-dimensional) computations.
Extending prior work by Hesse [52], we examine nonlinear Helmholtz and reaction-diffusion type test problems and particularly focus on examples that lead to a failure of simple shooting, necessitating the employment of a genuine multiple shooting method.

Different shooting approaches. In optimal control theory, there is a dichotomy of indirect and direct solution methods stemming from the underlying calculus of variations. Shooting methods for ODE control problems reflect this classification. Originally, multiple shooting for OCP was based on the maximum principle, which placed it among the indirect approaches. The direct variants developed later were able to cope with complex problem structures more efficiently and are widely accepted as a state-of-the-art solution approach. As this distinction deeply influenced our understanding of multiple shooting, we revisit the issue at several points throughout this work.
Our focus is on OCP governed by parabolic PDE, which are abstractly given as

$$\min_{(q,u)} J(q,u) \quad \text{subject to } e(q,u) = 0. \tag{1.1}$$

Almost all existing contributions for suchlike problems involving shooting type methods concentrate on direct variants. The first mentioning of indirect multiple shooting (IMS) we are aware of in the PDE context is the previously mentioned work of Hesse who developed an indirect approach opposing the existing direct methods. Due to its structural resemblance to multiple shooting for BVP, indirect shooting is intuitively comprehensible. In contrast,

the direct multiple shooting (DMS) method presented by Hesse and the classical one based on the work of Bock (see, e.g., [11–14]) in the ODE context are considerably different. In this thesis, we provide a thorough presentation of both IMS and DMS and we elaborate the connection between the different DMS formulations. Both direct and indirect shooting variants are identified as solvers for two different decompositions of the optimality system of the original OCP. Furthermore, the different DMS formulations can be interpreted as a reduced and a non-reduced approach for solving the same underlying decomposition of the optimality system, following an abstract perspective which is common in modern PDE optimization. These results have been submitted as a contribution to an anthology in the following article [21]:

T. Carraro, M. Geiger: *Direct and indirect multiple shooting for parabolic optimal control problems*, to appear in: Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015.

Control constrained problems. OCP solved by multiple shooting in the ODE framework are often complex and involve additional constraints on the control and/or state variables. Although in the past decade, much research effort was spent on the treatment of such constraints in the PDE optimization framework, no one has combined constrained parabolic OCP with multiple shooting yet. In this thesis, we design a suitable formulation of the constrained OCP enabling the use of shooting methods. We hereby concentrate on box constraints on the control, meaning that problem (1.1) is complemented by the condition

$$q_-(x,t) \leq q(x,t) \leq q_+(x,t) \tag{1.2}$$

almost everywhere in the space-time solution domain. Projected gradient and projected Newton methods are two widely used techniques coping with such constraints and we apply them to the multiple shooting problem formulation. Moreover, we combine the modern primal-dual active set strategies with the shooting algorithm. For the IMS case, the results achieved with these approaches have been published in the article [22]:

T. Carraro, M. Geiger, R. Rannacher: *Indirect multiple shooting for nonlinear parabolic optimal control problems with control constraints*, SIAM J. Sci. Comput. (36), 2014, pp. A452-A481.

Beyond the scope of the article, this thesis provides a comparison of IMS and DMS for both the unconstrained and the control constrained parabolic OCP framework.

Adaptivity. The original intention of extending simple shooting for BVP to multiple shooting by splitting the solution interval was to stabilize the solver. Common stability estimates in the ODE framework take the form

$$\|u(t;s_1) - u(t;s_2)\| \leq e^{t-t_0}\|s_1 - s_2\|, \tag{1.3}$$

meaning that a difference in the initial values s is exponentially increased over the time interval. Shortening the interval results in smaller stability constants. However, several numerical test examples, presented throughout this work, suggest that multiple shooting solvers are most efficient when only few shooting intervals are used. Particularly in the

PDE case where the space discretization must be accounted for, an increasing number of shooting intervals entails ever larger linear systems, which leads to a high computational effort due to a deteriorated conditioning. Thus, there is a trade-off between stability requiring as many shooting intervals as necessary and computational efficiency permitting as few shooting intervals as possible.
In order to find a suitable shooting grid for a given problem, we design an adaptive shooting approach. For temporal or spatial discretizations of ODE or PDE, adaptivity is by now a standard numerical feature that provides optimized time grids or spatial meshes. However, the existing literature contains almost no results on optimal shooting grids, not even for ODE based BVP and OCP.
Driven by the mentioned numerical examples, we develop two different extensions of the multiple shooting algorithm in the ODE context which achieve an automatic and problem-oriented choice of the respective shooting grids. A sensitivity-based technique which is inspired by an idea of Mattheij & Staarink [83] seeks appropriate shooting points in each shooting iteration without any prior knowledge. Although it yields adaptive splittings of the interval, the solution process is slowed down by a computational overhead required by the process.
As an alternative, we present a residual-based adaptive algorithm that starts from a given equidistant shooting grid and successively inserts (respectively removes) shooting points wherever necessary (respectively possible). It turns out that the number of shooting intervals originally prescribed is usually reduced during the iterative process.
Both adaptive mechanism are based on existing ideas but constitute original work. In particular, employment in the PDE context is novel. As the residual-based approach is more efficient in all considered ODE examples, we choose to only transfer this latter adaptive algorithm to the PDE framework. It is then applied to parabolic OCP, and finally also to PDE examples with additional control constraints.

Outline. We conclude this introduction presenting a short chapter-wise overview of the remainder of this thesis.

Chapter 2. We repeat the concept of shooting methods for boundary value problems in the ODE context. After a brief historical survey in Section 2.1, we present the basic algorithm as well as conditioning and stability issues in Section 2.2. Multiple shooting is embedded in the ODE optimal control framework in Section 2.3, where we introduce direct and indirect shooting methods. As a connection to the PDE context, multiple shooting is applied to nonstationary PDE initial boundary value problems (IBVP) in the final Section 2.4.

Chapter 3. We present OCP with parabolic PDE side conditions in their functional analytic context in Section 3.1. The required classical results on existence and uniqueness of solutions are recapitulated in Section 3.2, and optimality conditions and derivative generation are addressed in Section 3.3. The concluding Section 3.4 transfers the former results to the generalized situation of an extended OCP required in the multiple shooting context.

Chapter 4. In Section 4.1, the temporal and spatial discretization schemes employed in the practical implementation are briefly presented. Sections 4.2 and 4.3 contain introductions

to iterative solvers for linear systems, especially Krylov subspace methods, as well as to Newton type methods for nonlinear problems, respectively.

Chapter 5. This chapter formulates the direct and indirect shooting approaches in the nomenclature of modern PDE optimization. Both variants are closely connected on an abstract function space level. Sections 5.1 and 5.2 discuss IMS and DMS, respectively. The presented DMS method differs from the DMS approach common in ODE optimal control. Section 5.3 illustrates that the latter is based on a reduced formulation of the extended OCP and can be transferred into the DMS approach from Section 5.2. Further aspects of IMS and DMS are presented in Section 5.4. The final Section 5.5 substantiates the theoretical results by several numerical tests.

Chapter 6. The results of the previous chapter are extended to parabolic OCP with additional constraints. Exemplarily, box constraints on the control variable are considered. The OCP formulation is adapted in Section 6.1. Modern algorithms for constrained PDE optimization are tailored to the multiple shooting framework in Section 6.2, both for IMS and for DMS. The numerical tests displayed in Section 6.3 illustrate the theoretical results and enable an extended comparison of IMS and DMS in Section 6.4.

Chapter 7. This chapter deals with adaptivity in the multiple shooting context. As the examples from former chapters suggest, a problem-oriented choice of both number and position of subinterval endpoints enables a reduction of the computational effort. The literature lacks results on this topic even in the ODE context. A sensitivity-based approach to distributing the shooting points is presented in Section 7.1. This approach works only for linear BVP. We extend it to the nonlinear ODE case in Section 7.2. Furthermore, we develop a different, residual-based method for shooting grid adaptation. The transfer to the PDE framework raises additional difficulties, which result in our focussing on the residual-based adaptive shooting method. In Section 7.3 we apply this adaptive technique to parabolic OCP including control constrained problems. Each section contains several numerical tests illustrating the performance of the adaptive processes.

Chapter 8. The final chapter resumes the achieved results and develops ideas to extend our research presented in this thesis. This concerns the potential of shooting methods for parallel computing, state constraints for parabolic OCP in the shooting context, as well as further aspects of adaptivity.

2 Background of Multiple Shooting Methods

This chapter introduces different problem classes that can be treated by means of multiple shooting. After giving an overview on the historical development of shooting methods in Section 2.1, the subsequent sections cover boundary value problems (BVP) and optimal control problems (OCP) in the ODE context. We repeat the basic features of simple and multiple shooting in a context where these methods are established as standard solution routines. Our intention is to facilitate the understanding of the technically more complex multiple shooting methods in the PDE optimal control framework that are developed in Chapter 5. Both the theory and examples presented in Sections 2.2 and 2.3 raise the question of how to find good shooting grids for a given problem; this is answered in Chapter 7. We conclude the current chapter with a brief presentation of PDE initial boundary value problems (IBVP) in Section 2.4, illustrating how shooting methods can be applied in this context. This last section leads over to the PDE framework, as the remainder of this thesis is mainly concerned with parabolic OCP.

2.1 Brief historical overview

The origins of multiple shooting. First attempts to solve BVP in the ODE framework by shooting-like methods date back to the 1950s (see, e. g., Goodman & Lance [45]). The notion of multiple shooting was developed in an article by Morrison et al. [88]. Nievergelt [89] applied a similar procedure to an ODE initial value problem (IVP), aiming at solving the subinterval problems in parallel. By the 1970s, multiple shooting was well established as a BVP solver, see Osborne [92] or, later on, Keller [64].

Shooting methods for BVP. The development of shooting methods was originally motivated by BVP comprising a system of ODE and a set of boundary conditions that may both be nonlinear. Shooting methods for such problems turn the BVP into a sequence of simpler IVP. This transfer induces new problems. Even for well-conditioned BVP, the corresponding IVP are often ill-conditioned, and it is known for a wide range of BVP that simple shooting is unstable. The employment of multiple shooting, i. e., the decomposition of the solution interval into smaller subintervals, stabilizes the solution process.
From the early 1970s, Bulirsch and others provided important contributions to the analysis and application of shooting methods. Convergence of multiple shooting usually means convergence of Newton's method for the shooting system. The latter was studied in the multiple shooting context, e. g., by Weiss [114] who observed that increasing the number of

shooting intervals often enlarges the domain of suitable starting values for Newton's method. Both Deuflhard (see [32] or[33]) and Lory [77] suggested homotopy or continuation methods to enlarge the domain of convergence of Newton's method in the context of shooting techniques. The textbook by Deuflhard [34] gives a survey of general results on Newton type methods with special consideration of multiple shooting.
Further convergence results concern the convergence orders for the discrete subinterval solutions. If an IVP solver of order $\mathcal{O}(\Delta t^m)$ is used for linear BVP, then the shooting solution also converges of order $\mathcal{O}(\Delta t^m)$ (Δt denoting the timestep length). Nonlinear BVP are discussed in Jankowski [61] or Hieu [56], but general results do not exist and the topic is not treated in standard textbooks such as Ascher et al. [2].
In the 1980s, the interrelations between BVP conditioning and the stability of shooting methods were examined. First results were achieved by George & Gunderson [44], and the most important contributions originate from Mattheij and co-workers (see, e. g., [31, 70, 71, 81, 82]), who were mostly concerned with linear BVP. Mattheij's work is discussed in Section 2.2, and his approach to finding optimal shooting grids is extended in Chapter 7.

Shooting methods for ODE optimal control problems. Bulirsch [19] introduced shooting methods to the class of optimal control problems (OCP). His approach (see also [20]) is known as indirect multiple shooting (IMS). It is intuitive as it applies multiple shooting to the system of first order optimality conditions of such OCP. This optimality system constitutes the basis of most solution algorithms and is structurally similar to BVP. The IMS method is an example for a 'first-optimize-then-discretize' approach.
In the 1980s, an alternative shooting approach was developed which is known as direct multiple shooting (DMS). The most influential publications in this context originated from Bock and his co-workers (see, e.g., [11–15]). DMS is based on a discretization of the solution variables leading to a finite-dimensional optimization problem and is therefore a 'first-discretize-then-optimize' approach. The finite-dimensional problem is solved by suitable methods for nonlinear programming problems (NLP), e. g., sequential quadratic programming (SQP). A detailed description of DMS is given in Leineweber [73]. Further publications focussing on applications of DMS are Diehl et al. [35], Leineweber et al. [74] or Potschka [93].
Current research on ODE optimization concerns, e. g., the employment of multiple shooting in the field of ODE optimal experimental design (see Körkel et al. [69]). Furthermore, OCP with parabolic PDE constraints can be discretized by means of the method of lines (MOL), i. e., a spatial mesh is fixed before the temporal variable is discretized. The PDE is thus transcribed into a large ODE system, which requires algorithms for computing large-scale optimization problems (cf. the PhD theses of Albersmeyer [1] and Potschka [94]).
Details of IMS and DMS are discussed in Section 2.3 for ODE control problems and in Chapter 5 for PDE control problems. The classification into IMS and DMS reflects a general dichotomy of direct and indirect methods known from the calculus of variations (see, e. g., Dacorogna [29]).

Shooting methods for PDE governed OCP. The analytical and numerical study of PDE constrained optimization problems has experienced intense research in the past two decades. Theoretical foundations were laid earlier (cf. Lions [75]), but the numerical treatment was difficult due to lacking computing power and memory capacity. Control

problems with parabolic side conditions constitute the focus of this thesis. Up to now, shooting methods are rarely used in the PDE context. With the exception of the mentioned MOL approach toward parabolic OCP, only few contributions deal explicitly with multiple shooting.
Serban et al. (cf. [105]) proposed a structured adaptive mesh refinement (SAMR) approach toward space grid adaptation. Comas [26] and Heinkenschloss [50] studied preconditioners for parabolic optimization problems solved by time domain decomposition methods, thereby selecting multiple shooting as a representative example method. Hesse [52] first studied different multiple shooting techniques for PDE based problems. This raised one of our central questions, namely how IMS and DMS variants and different DMS formulations are interrelated (see Chapter 5). So far, the only IMS based publication in PDE optimal control is Hesse & Kanschat [53]. However, a thorough examination of the IMS method itself is omitted in favor of error estimation techniques and spatial mesh adaptation. The other mentioned publications in the PDE context deal exclusively with variants of DMS.

2.2 Shooting methods for boundary value problems (BVP)

Shooting methods reduce BVP to initial value problems, which enables the usage of IVP integrators in order to solve the more complex BVP. However, even for well-conditioned BVP this transfer often induces a strong sensitivity on the (parameterized) initial data and thus ill-conditioning of the corresponding IVP. This phenomenon was explained by Mattheij [82] and is briefly discussed below.
It is reasonable to first consider ODE BVP because the application of shooting methods to OCP is based on the fact that the first order necessary optimality conditions (the so-called Karush-Kuhn-Tucker or KKT system) can usually be interpreted as a coupled BVP with separated boundary conditions. Therefore, the class of BVP is briefly recapitulated, and a first variant of the multiple shooting algorithm is sketched. Despite the simplicity of the discussed problems, they bring up questions that are up to now not answered in a satisfactory way; their solution will be addressed in later chapters.
The type of BVP considered here is given by

$$\begin{aligned} \dot{u}(t) &= f(t, u(t)), \quad t \in [a, b], \\ 0 &= r(u(a), u(b)). \end{aligned} \tag{2.1}$$

Remark 2.1. Whenever referring to ODE problems, we denote differentiation of the solution function $u(t)$ w. r. t. the time variable t by an overdot, i. e. $\frac{\mathrm{d}}{\mathrm{d}t}u(t) =: \dot{u}(t)$, whereas differentiation for a parameter, e. g. $\frac{\mathrm{d}}{\mathrm{d}s}u(t; s)$, is denoted by $u'_s(t; s)$.

The function $u \in C^1[a, b]^d, d \geq 1$, denoted in (2.1) should fulfil both the differential equation and the boundary condition. Furthermore, both $f(t, \cdot)$ and $r(\cdot, \cdot)$ may be nonlinear and are assumed to be at least twice continuously differentiable in each component on the time interval $I = [a, b]$.

Remark 2.2. Equation (2.1) describes the general BVP. An important subclass of BVP comprises problems where both the differential equation and the boundary conditions are linear:

$$\begin{aligned} \dot{u}(t) &= A(t)u(t) + b(t), \quad t \in [a,b], \\ 0 &= B_a u(a) + B_b u(b) - g. \end{aligned} \tag{2.2}$$

Here, $A(\cdot) : I \to \mathbb{R}^{d\times d}$ and $b(\cdot) : I \to \mathbb{R}^d$ are continuous real-valued matrix and vector functions, respectively, $B_a,\ B_b \in \mathbb{R}^{d\times d}$ are given constant matrices, and $g \in \mathbb{R}^d$ is a given vector. We note that, even if the differential equation in problem (2.1) is nonlinear, often the imposed boundary conditions are linear as in the second equation of (2.2).

The idea behind simple shooting is to replace the initial value $u(a)$ by a parameter s and solve the initial value problem

$$\begin{aligned} \dot{u}(t) &= f(t,u(t)), \quad t \in [a,b], \\ u(a) &= s, \end{aligned} \tag{2.3}$$

thereby computing s in a way that the right boundary value $u(b;s)$ is matched. We obtain a solution $u(t;s)$ to the underlying BVP if the additional condition

$$r(s,u(b;s)) = 0 \tag{2.4}$$

is fulfilled. The determination of s can for instance be carried out by applying Newton's method to the function $F(s) := r(s,u(b;s))$, i. e. by iteratively solving the system

$$s_{i+1} = s_i - [F_s'(s_i)]^{-1} F(s_i), \tag{2.5}$$

where, in general, the starting point s_0 has to be chosen carefully in order to guarantee convergence. For this purpose, we have to compute the derivative (obtained by means of the chain rule)

$$F_s'(s) = r_x'(s,u(b;s)) + r_y'(s,u(b;s))u_s'(b;s). \tag{2.6}$$

Here, differentiating $r(\cdot,\cdot)$ w. r. t. its arguments poses no problem, but the computation of $u_s'(b;s)$ involves the solution of an additional linearized IVP, the so-called variational or sensitivity equation (for a detailed presentation and proof, see, e. g., Coddington & Levinson [25]):

$$\begin{aligned} \dot{G}(t;s) &= f_x'(t,u(t;s))G(t;s), \quad t \in [a,b], \\ G(a;s) &= I_d. \end{aligned} \tag{2.7}$$

This problem constitutes a matrix ODE, where $G(t;s) := u_s'(t;s)$, and I_d denotes the $d \times d$ identity matrix. The repetitive process of solving the IVP (2.3) with initial value s_i, evaluating $F(s_i)$, then solving the variational equation (2.7), evaluating $F_s'(s_i)$, and iterating this process until $\|F(s_i)\|_2$ falls below a given tolerance constitutes the simple shooting algorithm for ODE BVP.
This method has a severe drawback being displayed in an example further below: Even for linear and autonomous BVP (2.2), the corresponding IVP (2.3) may react sensitively to small perturbations in the boundary data, especially for long time intervals. This is

caused by a dichotomy relying on a splitting of the fundamental solution of problem 2.2 into exponentially increasing and decreasing components that are controlled by boundary conditions at the right and left solution interval endpoint, respectively (see Mattheij and co-workers, e. g., [31], [70] or [82]). Here, the fundamental solution is a $d \times d$ matrix function $\Phi(t)$ solving the ODE problem

$$\dot{\Phi}(t) = A(t)\Phi(t), \tag{2.8}$$

which describes the homogeneous part of (2.2) and, in the linear case, corresponds to the variational differential equation (2.7). The following considerations concern the BVP and are not linked to a specified solution routine such as simple or multiple shooting. Let $(\Phi^k(t))_{k=1}^d$ be the columns of the fundamental solution $\Phi(t)$ (which is uniquely determined only up to additional initial values at the left interval endpoint a). Assume further that i, j are integers with $0 \leq i + j \leq d$, that $\lambda > 0$ and $\mu < 0$ are real numbers and that c_1, c_2 and c_3 are positive constants, so that for all $t_1, t_2 \in [a, b]$ with $t_1 \leq t_2$ the following relations hold (where $\|\cdot\|$ is a natural matrix norm):

$$\begin{aligned} \|\Phi^k(t_1)\| &\leq c_1 e^{-\lambda(t_2-t_1)} \|\Phi^k(t_2)\| \qquad (k \leq j), \\ \|\Phi^k(t_2)\| &\leq c_2 e^{\mu(t_2-t_1)} \|\Phi^k(t_1)\| \qquad (k \geq n-i+1), \\ \|\Phi^k(t_2)\| &\leq c_3 \|\Phi^k(t_1)\| \qquad (j+1 \leq k \leq n-i). \end{aligned} \tag{2.9}$$

The first relation describes exponentially increasing solution components, the second one components that decrease exponentially, and the third one comprises essentially constant components.

Remark 2.3. In the autonomous case, i. e., $A(t) \equiv A$, the number j counts those eigenvalues with positive real part, whereas i corresponds to the number of eigenvalues with negative real part.

This classification of solution components is known as the dichotomic structure of linear BVP. It was first described in the 1970s (see, e. g., the article and book by Coppel [27], [28]), but only Mattheij clarified the connection between this dichotomy, the prescribed boundary conditions and the conditioning of the BVP in [82]. We now explain these interrelations and illustrate them further below by means of an example.
As initial value for the variational ODE (2.8), $\Phi(a) \equiv I$ (the $d \times d$ identity matrix) is chosen, which is common in the context of shooting methods (note, however, the scaling of the fundamental solution postulated below). The conditioning of a problem class is often described by so-called condition numbers (e. g., $\mathrm{cond}_2(A)$ is the spectral condition number of a linear equation system with matrix A). Mattheij proposed, based both on examples and on theoretical considerations, the following definition of a BVP condition number:

$$\mathrm{cond}_{BVP} := \max_{t\in[a,b]} \|\Phi(t)[B_a\Phi(a) + B_b\Phi(b)]^{-1}\| = \max_{t\in[a,b]} \|\Phi(t)[B_a + B_b\Phi(b)]^{-1}\|. \tag{2.10}$$

It is well-known that the BVP (2.2) is uniquely solvable if and only if the matrix $Q := B_a + B_b\Phi(b)$ is regular, which is also crucial for the definition of cond_{BVP}. As the quantity (2.10) is difficult to handle, Mattheij suggested a further estimate:

$$\mathrm{cond}_{BVP} \leq \max_{t\in[a,b]} \|\Phi(t)\| \|[B_a + B_b\Phi(b)]^{-1}\|. \tag{2.11}$$

In order to avoid the dependence on the fundamental solution $\Phi(t)$ (which is not uniquely determined), he proposed to scale the fundamental solution so that $\max_{t\in[a,b]} \|\Phi(t)\| = 1$, neglect the factor and consider the approximation

$$\mathrm{cond}_{BVP} \approx \kappa := \|[B_a + B_b\Phi(b)]^{-1}\|. \tag{2.12}$$

Mattheij was able to show that, with a function $c(t)$ based on a further diagonal scaling of $\Phi(t)$, the actual condition number can be estimated as follows:

$$c(t)\kappa \leq \mathrm{cond}_{BVP} \leq \kappa. \tag{2.13}$$

Based on this quantity κ, Mattheij called a BVP well-conditioned if $\kappa = \mathcal{O}(1)$ and ill-conditioned if $\kappa = o(1)$, where $\mathcal{O}(\cdot)$ and $o(\cdot)$ are the Landau functions. In this thesis we call a BVP, differently from Mattheij, well-conditioned in case that $\kappa \approx 1$, and ill-conditioned if $\kappa \gg 1$.
Mattheij was able to prove several statements on ill-conditioning of BVP, see [82]. His results suggest that, in terms of (2.9), exponentially increasing solution components should be controlled at the right interval boundary, whereas exponentially decreasing ones require conditions at the left interval boundary. This dichotomic structure is usually violated by turning the problem into an IVP, which causes ill-conditioning and entails a highly sensitive dependence on perturbations in the initial values. The latter is grounded in stability estimates of the form

$$\|u(t; s_1) - u(t; s_2)\|_2 \leq e^{L(t-a)}\|s_1 - s_2\|_2, \tag{2.14}$$

which show that the error in the initial value s affects the solution exponentially with increasing time t (L is the Lipschitz constant of the righthand side function $f(t,\cdot)$). This phenomenon renders the application of simple shooting practically impossible in many problem configurations. However, this drawback may be overcome by solving the IVP on shorter time intervals, which is the main idea of the multiple shooting method.

Remark 2.4. It is important to distinguish whether a given problem itself is ill-conditioned, which naturally leads to instabilities in the simple shooting algorithm, or whether these instabilities are only induced by using an IVP method for the possibly well-conditioned BVP, thus disturbing the dichotomic structure.

The first step toward multiple shooting is a decomposition

$$I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1} (\tau_j, \tau_{j+1}], \quad a =: \tau_0 < \tau_1 < \cdots < \tau_M := b \tag{2.15}$$

of the time domain I into smaller subintervals $I_j := (\tau_j, \tau_{j+1}]$ (in the following called shooting intervals). Next, we impose parameters s^j as artificial initial values at the time-points τ_j (denoted as shooting points), which results in the following set of IVP (for $j = 0, \ldots, M-1$):

$$\begin{aligned} \dot{u}^j(t) &= f(t, u^j(t)), \;\; t \in I_j, \\ u^j(\tau_j) &= s^j \end{aligned} \tag{2.16}$$

Due to the arbitrarily chosen shooting variables s^j, we cannot expect to obtain a globally continuous solution from the intervalwise IVP solutions. Instead, jumps will occur at the shooting points, which contradicts the above request for a globally continuously differentiable solution $u(t)$ to our BVP. Therefore, we have to assure that the mentioned jumps vanish. This can be done in a way similar to fulfilling the boundary conditions in (2.4). We have to introduce a system of matching conditions, forcing the boundary condition as well as the jumps to converge simultaneously to zero by applying Newton's method. These continuity conditions are given as follows:

$$\begin{aligned} u^j(\tau_{j+1}; s^j) - s^{j+1} &= 0, \quad j = 0, \dots, M-2, \\ r(s^0, u^{M-1}(\tau_M; s^{M-1})) &= 0. \end{aligned} \tag{2.17}$$

By defining a vector $\bar{s} := (s^0, s^1, \dots, s^{M-1})^\top$, we can abbreviate equations (2.17) and simply write $F(\bar{s}) = 0$, analogously to equation (2.4), and can again apply Newton's method to solve the matching conditions. As before, derivatives $G^j(t; s^j) := u^{j\prime}_{s^j}(t; s^j)$ of the intervalwise solution functions w. r. t. their respective initial values s^j are required, being obtained by solving intervalwise variational equations

$$\begin{aligned} \dot{G}^j(t; s^j) &= f'_x(t, u^j(t; s^j)) G^j(t; s^j), \quad t \in I_j, \\ G^j(\tau_j; s^j) &= I. \end{aligned} \tag{2.18}$$

The Jacobian of the system of matching conditions is given as

$$F'_s(\bar{s}_k) = \begin{pmatrix} G^0(\tau_1) & -I & 0 & \cdots & 0 \\ 0 & G^1(\tau_2) & -I & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & G^{M-2}(\tau_{M-1}) & -I \\ A & 0 & \cdots & 0 & B \end{pmatrix}, \tag{2.19}$$

with the matrices $A = r'_x(s^0, u^{M-1}(\tau_M; s^{M-1}))$ and $B = r'_y(s^0, u^{M-1}(\tau_M; s^{M-1})) G^{M-1}(\tau_M)$ in the last row.
At this point, the multiple shooting procedure has already become rather complex, and in order to keep track, we formulate it as a whole in Algorithm 2.1.

Algorithm 2.1 Multiple shooting for nonlinear boundary value problems

Require: Decomposition $I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1} (\tau_j, \tau_{j+1}]$, shooting variables $\bar{s}_0$
1: Set $k = 0$, prescribe tolerance TOL
2: **while** $\|F(\bar{s}_k)\|_2 > TOL$ **do**
3: Solve initial value problems (2.16), evaluate residual $-F(\bar{s}_k)$ computing (2.17)
4: Solve variational initial value problems (2.18), evaluate $F'_s(\bar{s}_k)$ as given in (2.19)
5: Solve shooting system $F'_s(\bar{s}_k)\delta\bar{s}_k = -F(\bar{s}_k)$
6: Compute update $\bar{s}_{k+1} = \bar{s}_k + \delta\bar{s}_k$, set $k \leftarrow k + 1$
7: **end while**

Remark 2.5. The multiple shooting method for linear BVP as derived in detail in the textbook of Bulirsch & Stoer [20] is a special case of the above Algorithm 2.1. In the linear case, Newton's method converges in one iteration step, and convergence is largely independent of the starting value $\bar{s}_0$, which is therefore chosen as $\bar{s}_0 \equiv 0$ for the sake of convenience.

The following example illustrates most of the features of shooting methods mentioned so far. Due to its simplicity, this example is an appropriate test case for several aspects of our work, and we will repeatedly reconsider variants of this example, especially in Chapter 7 (cf. also Remark 2.6 at the end of this section). The implementation has been carried out in `MATLAB`.

Example 2.1. *Consider the (fully linear) BVP*

$$\begin{pmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ c & 1 \end{pmatrix} \begin{pmatrix} u^1(t) \\ u^2(t) \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^1(0) \\ u^2(0) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1(10) \\ u^2(10) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

This BVP is defined on $I = [0, 10]$ and depends on a parameter c. The exact solution — with constants $a = \frac{1}{2} - \sqrt{c + \frac{1}{4}}$ and $b = \frac{1}{2} + \sqrt{c + \frac{1}{4}}$ — is obtained as

$$u_1(t) = \frac{e^{10b} - 1}{e^{10b} - e^{10a}} e^{at} + \frac{1 - e^{10a}}{e^{10b} - e^{10a}} e^{bt}, \quad u_2(t) = a \frac{e^{10b} - 1}{e^{10b} - e^{10a}} e^{at} + b \frac{1 - e^{10a}}{e^{10b} - e^{10a}} e^{bt}.$$

This problem is (for large c) very sensitive to perturbations of the boundary values. For $c = 110$ and the prescribed boundary conditions, $u_1(0) = 1$ and $u_1(10) = 1$, the exact initial value in the second component is $u_2(0) \approx -10 + 3.5 \cdot 10^{-47}$. A perturbation to $u_2(0) = -10 + 10^{-9}$ (i. e. a perturbation of size $\approx 10^{-9}$) leads to the value $u_1(10) \approx 10^{37}$. The above theory yields two reasons why this problem cannot be solved in a straightforward way. First, a corresponding fundamental solution is given by

$$\Phi(t) = \begin{pmatrix} e^{11t} & e^{-10t} \\ 11e^{11t} & -10e^{-10t} \end{pmatrix}$$

where $e \approx 2.71828$ is the Euler constant. Thus, the matrix $Q := B_0 + B_{10}\Phi(10)$ and its inverse are given by

$$Q = \begin{pmatrix} 1 & 0 \\ e^{110} & e^{-100} \end{pmatrix}, \quad Q^{-1} = \begin{pmatrix} 1 & 0 \\ -e^{210} & e^{100} \end{pmatrix}.$$

In this case, the condition number cond_{BVP} defined in (2.10), measured in the $\|\cdot\|_\infty$ norm, is $\mathrm{cond}_{BVP} \approx 5.9 \cdot 10^{47}$, and the approximate condition number κ from (2.12) is given by $\kappa \approx 1.6 \cdot 10^{91}$. Even scaling the fundamental solution so that $\max_{t \in [a,b]} \|\Phi(t)\| \approx 1$ still yields $\kappa \approx 3.0 \cdot 10^{44}$. This renders the problem extremely ill-conditioned. Second, the problem itself has one exponentially increasing solution component, $u_1 = c_1 e^{11t}$, and one exponentially decreasing solution component, $u_2 = c_2 e^{-10t}$. The latter should be controlled at the right interval boundary $b = 10$, which is not the case in the problem configuration;

Table 2.1. Example 2.1: The minimum number of shooting intervals (SI) needed for solving the BVP as a function of the parameter c. Data: $TOL = 10^{-8}$, $8! = 5040$ timesteps (ts) equally distributed to the SI.

c	#SI	#ts/SI	$\|F(\bar{s}_0)\|_2$	$\|F(\bar{s}_1)\|_2$
1	1	5040	$7.7 \cdot 10^5$	$3.1 \cdot 10^{-9}$
2	2	2520	$3.6 \cdot 10^3$	$6.1 \cdot 10^{-11}$
10	3	1680	$7.3 \cdot 10^4$	$3.0 \cdot 10^{-10}$
22	4	1260	$2.4 \cdot 10^5$	$1.7 \cdot 10^{-9}$
43	5	1008	$1.1 \cdot 10^6$	$3.7 \cdot 10^{-9}$
67	6	840	$2.1 \cdot 10^6$	$2.5 \cdot 10^{-9}$
95	7	720	$3.1 \cdot 10^6$	$3.8 \cdot 10^{-9}$
110	8	630	$1.4 \cdot 10^6$	$4.1 \cdot 10^{-9}$

therefore, the boundary conditions do not fit the dichotomic structure of the problem. Table 2.1 states that increasing the value of the parameter c necessitates an increasing amount of (equidistantly distributed) shooting intervals; here, the selected values of c mark thresholds where the (minimum) number of shooting intervals must be incremented. Figure 2.1 displays the computed solution for $c = 110$ and $M = 8$ shooting intervals. Before convergence, the typical exponential growth of the single solution arcs can be observed, whereas after convergence a globally continuous solution is obtained.
Furthermore, the minimum number of shooting intervals to solve the above problem for a

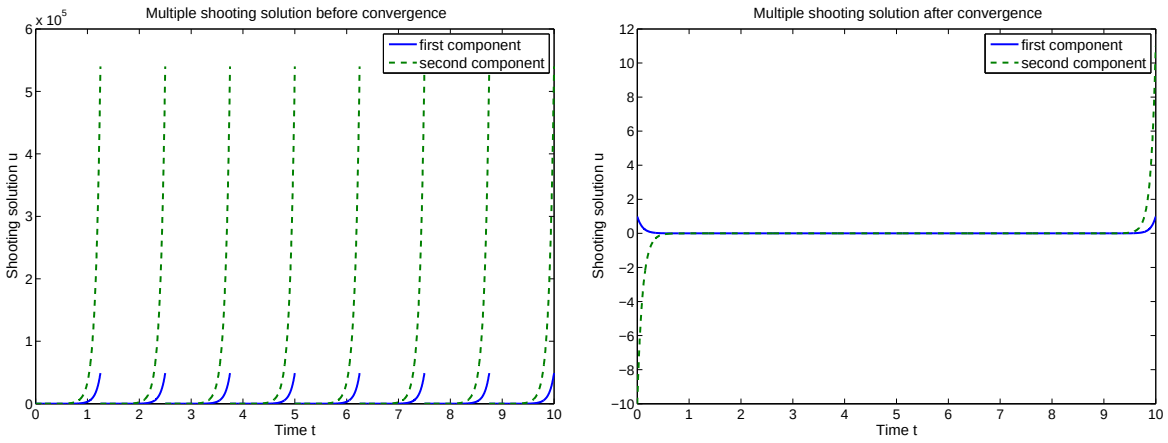

Figure 2.1. Example 2.1: Multiple shooting solution before (left) and after (right) convergence.

given fixed value of c can be read from Table 2.1. Indeed, one can choose more shooting intervals than indicated in the table (up to the extreme case where the distribution of shooting intervals matches the time discretization, i. e. there is only one time step per shooting interval). Increasing the number of shooting intervals at first renders the problem more stable (due to estimates such as (2.14)). As a measure for this, the condition number of

the solution $G(t)$ of the variational equation (2.18) is considered. We choose the maximum value of the spectral condition on the solution interval I, i. e., $\max_t \mathrm{cond}_2(G(t))$, for Table 2.2 and Figure 2.2, thus preparing further work presented in Chapter 7. However, using too many shooting intervals leads to higher computational costs, as the Newton system corresponding to (2.5) grows larger and its condition number increases (again, we choose the spectral condition $\mathrm{cond}_2(F'(\overline{s}))$).
In Table 2.2, we observe this trade-off between local and global conditioning. The increase

Table 2.2. Example 2.1: The maximum local condition number of the solution $G(t)$ of the variational equation, the global condition number of $F_s'(\overline{s})$, and the computing time (in seconds); comparison of the parameter values $c = 1$ and $c = 10$.

		$c = 1$			$c = 10$		
#SI	#ts/SI	$\mathrm{cond}_2(G)$	$\mathrm{cond}_2(F')$	time(s)	$\mathrm{cond}_2(G)$	$\mathrm{cond}_2(F')$	time(s)
3	1680	$1.73 \cdot 10^3$	$2.67 \cdot 10^2$	1.93	$5.53 \cdot 10^9$	$1.13 \cdot 10^6$	1.65
6	840	$4.15 \cdot 10^1$	$2.34 \cdot 10^1$	1.94	$1.28 \cdot 10^5$	$2.38 \cdot 10^3$	1.66
10	504	$9.36 \cdot 10^0$	$1.22 \cdot 10^1$	1.93	$1.79 \cdot 10^3$	$2.04 \cdot 10^2$	1.67
20	252	$3.06 \cdot 10^0$	$1.13 \cdot 10^1$	1.95	$6.93 \cdot 10^1$	$3.60 \cdot 10^1$	1.65
40	126	$1.75 \cdot 10^0$	$1.62 \cdot 10^1$	1.97	$1.13 \cdot 10^1$	$2.10 \cdot 10^1$	1.69
80	63	$1.32 \cdot 10^0$	$2.78 \cdot 10^1$	1.97	$3.74 \cdot 10^0$	$2.47 \cdot 10^1$	1.67
630	8	$1.04 \cdot 10^0$	$1.94 \cdot 10^2$	2.12	$1.19 \cdot 10^0$	$1.35 \cdot 10^2$	1.84
1260	4	$1.02 \cdot 10^0$	$3.84 \cdot 10^2$	2.96	$1.09 \cdot 10^0$	$2.64 \cdot 10^2$	2.67

in the global condition number entails higher computing times. In this simple example, the growth of memory requirement and computing time is negligible, but we will encounter examples in the context of PDE governed optimal control where an increase in computing time by a factor of two already has a large impact (see Chapter 5).

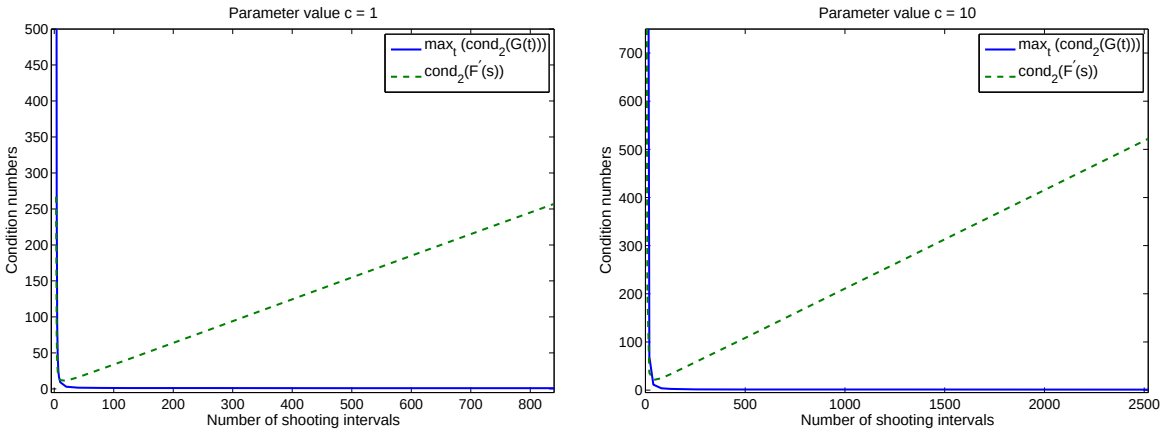

Figure 2.2. Example 2.1: Local and global condition numbers for $c = 1$ (left) and $c = 10$ (right); linear growth of the conditioning of the Newton matrix.

Remark 2.6. Example 2.1 suggests that a criterion allowing to determine the number of shooting intervals (as well as possibly their distribution in the solution interval) for a given problem in advance might be desirable. This becomes even more worthwile if the parameter c is time-dependent and takes on large values within a short subinterval, but is otherwise small. In the context of linear ODE BVP, Mattheij and his co-workers (see [82] or [83]) proposed a method to achieve this. Their approach will be presented in detail in Chapter 7, where we discuss its drawbacks and try to overcome them. Subsequently it will be extended to the nonlinear case, and both cases will also be transferred to the PDE context.

2.3 Ordinary differential equations (ODE) based control problems

Next, ODE optimal control problems (OCP) of the following form are considered on a finite solution interval $I = [t_0, t_f]$:

$$\min_{(q,u)} J(q,u) \quad \text{s. t. } F(t; u(t), \dot{u}(t), q(t)) = 0. \tag{2.20}$$

The general notation for OCP will be explained in detail in Chapter 3 where all theoretical results on optimal control required in this work are collected. The abstract framework presented there comprises both ODE and parabolic PDE control problems. Here, we constrain ourselves to the following brief explanations. In (2.20), the cost (or objective) functional $J(q,u)$ is the quantity to be minimized given by

$$J(q,u) = \frac{1}{2} \int_I [\Phi(t, u(t)) + \Psi(q(t))] \, \mathrm{d}t, \tag{2.21}$$

where Φ is assumed as a tracking-type function $\|u(t) - \bar{u}(t)\|^2$ of the state variable u, and the regularization term $\Psi(q(t))$ as $\alpha\|q(t) - \bar{q}(t)\|^2$ with q denoting the control variable. Here, $\bar{u}$ and $\bar{q}$ are given functions to be fitted as close as possible in the $L\check{s}$ sense through the optimization process. $F(t; u, \dot{u}, q)$ in (2.20) is called the side condition and may be a nonlinear ODE system. It is commonly concretized as an explicit ODE system defined on $I = [t_0, t_f]$, and in the current section only the following special case of an IVP with linear dependence on q is treated:

$$\dot{u}(t) = f(t, u(t)) + cq(t), \quad u(t_0) = u_0. \tag{2.22}$$

This simple problem may be generalized in several ways; Leineweber [73] provides a summary of different generalizations w. r. t. functional types and side conditions, additional control or state constraints, or problems depending on further parameters. As the goal of this section is to prepare the tools for the PDE optimal control case, we do not cover the most general ODE examples. Instead, OCP of the special type (2.21) – (2.22) are used in order to explain the basic difference between direct (DMS) and indirect multiple shooting (IMS) methods. These two classes of shooting techniques appear only in the

optimal control framework and mirror the general dichotomy between direct and indirect solution methods for control problems. We return to ODE governed OCP in Chapter 7 where strategies for choosing the shooting intervals adaptively are provided in the ODE case before transferring them to the more complicated PDE framework.
Subsection 2.3.1 clarifies the relation between BVP and OCP via the Lagrange functional $\mathcal{L}(q, u, z)$ and the system of first order optimality conditions (KKT system) consisting of the derivatives of $\mathcal{L}$. This leads to the formulation of IMS. Afterwards, DMS is derived in Subsection 2.3.2 following a different approach. However, both shooting variants are closely connected, which is shown in Chapter 5 in the PDE context.
The employment of shooting methods in ODE optimal control is still a research topic despite three decades of experience. Some related areas where these methods are in the focus of current developments are, e. g., the field of optimal experimental design aiming at optimizing expensive experiment configurations in real world applications and the industrial sector by conducting accurate simulations (see, e. g., Körkel et al. [69]), or parameter estimation which becomes more and more important in biological applications and where multiple shooting is efficiently applied (cf. Bock et al. [16]). Other recent applications, especially of direct shooting variants, can be found in Diehl et al. [35] or Leineweber et al. [74]. Furthermore, the treatment of PDE problems by ODE methods in the method of lines (MOL) framework is a subject of current research (see, e. g., Albersmeyer [1] or Potschka [94]) to which we refer back in Section 2.4.

2.3.1 Indirect approach

The IMS method for the above OCP resembles the shooting method originally developed for ODE boundary value problems. We therefore transform the above problem (2.21)–(2.22) into a BVP by differentiating the Lagrange functional

$$\mathcal{L}(q, u, z) := J(q, u) + \int_I (\dot{u}(t) - f(t, u(t)) - cq(t), z(t))\,\mathrm{d}t + (u(t_0) - u_0, z(t_0)) \tag{2.23}$$

with respect to its arguments, where the Lagrange multiplier z acts as an adjoint variable. The derivatives of $\mathcal{L}(q, u, z)$ form the first order necessary optimality conditions (the so-called Karush-Kuhn-Tucker or KKT system). With $\xi := (q, u, z)$, they read:

$$\mathcal{L}'_z(\xi)(\varphi) = \int_I (\dot{u}(t) - f(t, u(t)) - cq(t), \varphi(t))\,\mathrm{d}t + (u(t_0) - u_0, \varphi(t_0)) \stackrel{!}{=} 0, \tag{2.17a}$$

$$\mathcal{L}'_u(\xi)(\psi) = J'_u(q, u)(\psi) - \int_I (\dot{z}(t) + f'_u(t, u(t)), \psi(t))\,\mathrm{d}t + (z(t_f), \psi(t_f)) \stackrel{!}{=} 0, \tag{2.17b}$$

$$\mathcal{L}'_q(\xi)(\chi) = J'_q(q, u)(\chi) - \int_I (cz(t), \chi(t))\,\mathrm{d}t \stackrel{!}{=} 0. \tag{2.17c}$$

Each stationary point of these KKT conditions is a solution candidate for the original control problem. With $\Psi(q(t)) = \alpha\|q(t)\|^2$ from (2.21), one can reformulate the control or

gradient equation (2.17c) as follows:

$$\int_I (\alpha q(t) - cz(t), \chi(t))\,\mathrm{d}t \stackrel{!}{=} 0.$$

Dividing by the regularization parameter $\alpha > 0$ yields an expression for q in terms of the adjoint variable z:

$$\int_I (q(t), \chi(t))\,\mathrm{d}t = \frac{1}{\alpha}\int_I (cz(t), \chi(t))\,\mathrm{d}t. \tag{2.25}$$

In addition, the state u, adjoint z and control q are assumed to be elements of the same function space, i. e. the test functions φ, ψ and χ in (2.17a)–(2.17c) can also be chosen from the same space. In the ODE case where it holds $u, z \in C^1(I) \subset C^0(I)$, this is not restrictive, and the assumption $q \in C^0(I)$ is justified; on the discrete level, we can discretize q analogously to u and z (see Chapter 4). Therefore, $q(t)$ in (2.17a) can be replaced by the expression $\frac{c}{\alpha}z(t)$ from (2.25), and we obtain the following system (where, for brevity, the argument t in the integral terms is neglected):

$$\int_I \left(\dot{u} - f(t,u) - \frac{c^2}{\alpha}z, \varphi\right)\mathrm{d}t + (u(t_0) - u_0, \varphi(t_0)) = 0, \tag{2.17a*}$$

$$J_u'(q,u)(\psi) - \int_I (\dot{z} + f_u'(t,u), \psi)\,\mathrm{d}t + (z(t_f), \psi(t_f)) = 0. \tag{2.17b*}$$

These two equations constitute an ODE boundary value problem (BVP) with separated boundary conditions $u(t_0) = u_0$ and $z(t_f) = 0$. Here, the adjoint equation runs backward in time, which can be seen from the negative sign of the time derivative as well as the initial condition prescribed at the final time point t_f.

Remark 2.7. The cost functional $J(q,u)$ contains a tracking-type term for the state u. An alternative term $\phi(u(t_f))$ at the final time-point would result in a different boundary condition in the adjoint equation (2.17b*), i. e., $z(t_f) \neq 0$.

Remark 2.8. The above assumption (u, z and q contained in the same space) enables a reduction of the system of solution variables to u and z, whereas q may simply be evaluated via (2.25). It is already quite restrictive in the ODE case, where u and z have to be at least in $C^1(I)$, whereas q often is not even required to be continuous (as an example, consider optimization problems with bang-bang control). Thus, the above decoupling of the control cannot be performed in general. Here, the focus is on problems where the described replacements are possible, but later on in the PDE context, we will introduce an alternative approach and thus avoid this system reduction to u and z.

While the weak formulation (2.17a*)–(2.17b*) of the problem constitutes the natural environment for problems that are solved by variational methods, the BVP structure becomes more obvious from the strong formulation

$$\dot{u} = f(t,u) + \frac{c^2}{\alpha}z, \qquad u(t_0) = u_0, \tag{2.17a**}$$

$$\dot{z} = -f_u'(t,u) - j_u'(q,u), \qquad z(t_f) = 0, \tag{2.17b**}$$

where $j'_u(q,u)$ represents the strong form of the term $J'_u(q,u)(\psi)$ from (2.17b*) (in case of a tracking-type term, this means $j'_u(q,u) = u - \overline{u}$ with a tracking function $\overline{u}$).
We now replace the boundary value at the final time by an additional parameterized initial value $z(t_0) = s$, solve both components of problem (2.17a**)–(2.17b**) forward in time and end up with the additional nonlinear equation $z(t_f; s) = 0$ to be solved at the final time t_f. By this, the BVP is transformed into an IVP, which is the aim of the simple shooting algorithm introduced in the previous section. However, it has already been shown that this method is often very sensitive to perturbations in the data and thus constitutes an unstable algorithm. We therefore proceed to the multiple shooting technique, which overcomes the mentioned drawback and has improved stability properties. This relies again upon the interval decomposition (2.15) with the intermediate points $t_0 = \tau_0 < \tau_1 < \cdots < \tau_M = t_f$. In this way, the local exponential stability factors $e^{L(\tau_{j+1}-\tau_j)}$ in (2.14) are of moderate size and the problem is stabilized (see Example 2.1). However, the BVP (2.17a**)–(2.17b**) has to be solved locally on the subintervals I_j, which requires adequate initial values on each subinterval. As the exact solution values in τ_j are unknown and there usually is no appropriate starting value, we have to prescribe artificial parameterized values $s^j = (s_u^j, s_z^j)$. To receive a uniform algorithm on all subintervals, therefore the value u_0 has to be replaced by a parameter s_u^0. The local solutions $(u^j(t; s^j), z^j(t; s^j))$ depend on the erroneous initial values and yield incorrect values $(u^j(\tau_{j+1}; s^j), z^j(\tau_{j+1}; s^j))$. This induces jumps in the global solution at the points τ_j which have to be removed in order to obtain the correct globally continuous solution of the original problem. In the case of multiple shooting, we thus have to fulfil the boundary conditions as well as additional matching conditions, together forming the following system:

$$\begin{aligned} s_u^0 - u_0 &\stackrel{!}{=} 0, \\ s_u^{j+1} - u^j(\tau_{j+1}; s_u^j, s_z^j) &\stackrel{!}{=} 0, \quad (j = 0, \cdots, M-1) \\ s_z^{j+1} - z^j(\tau_{j+1}; s_u^j, s_z^j) &\stackrel{!}{=} 0, \quad (j = 0, \cdots, M-1) \\ z^{M-1}(\tau_M; s_u^{M-1}, s_z^{M-1}) &\stackrel{!}{=} 0 \end{aligned} \tag{2.26}$$

(compare this to (2.17) in Section 2.2). This system is abbreviated by $F(s) \stackrel{!}{=} 0$ where $s = ((s_u^j, s_z^j)_{j=0}^{M-1})$, and for solving this equation, Newton's method is employed. Therefore, we need the derivative $F'_s(s)$, and thus the local derivatives $u^{j\prime}_{s_u^j}, u^{j\prime}_{s_z^j}, z^{j\prime}_{s_u^j}$ and $z^{j\prime}_{s_z^j}$, each evaluated in τ_{j+1}. The simplest way to get these derivatives is to solve additional local problems on the subintervals I_j, the so-called variational or sensitivity equations obtained by linearizing the local system corresponding to (2.17a**)–(2.17b**) w. r. t. (s_u^j, s_z^j). Thus, the sensitivity equation consists of a matrix ODE

$$\begin{pmatrix} \dot{u}^{j\prime}_{s_u^j} & \dot{u}^{j\prime}_{s_z^j} \\ \dot{z}^{j\prime}_{s_u^j} & \dot{z}^{j\prime}_{s_z^j} \end{pmatrix} = \begin{pmatrix} f'_u(t, u^j) & \frac{c^2}{\alpha} \\ -f''_{uu}(t, u^j) - j''_{uu}(q^j, u^j) & 0 \end{pmatrix} \begin{pmatrix} u^{j\prime}_{s_u^j} & u^{j\prime}_{s_z^j} \\ z^{j\prime}_{s_u^j} & z^{j\prime}_{s_z^j} \end{pmatrix}, \tag{2.27}$$

and the initial value on each subinterval is given by the identity matrix of corresponding dimension. This matrix ODE corresponds to (2.18). Taking the results together, the indirect multiple shooting algorithm (Algorithm 2.2) for problem (2.21)–(2.22) can now be formulated.

Algorithm 2.2 Indirect multiple shooting for ODE governed OCP

Require: Initial control $q_0 = ((q_0^j)_{j=0}^{M-1})$, decomposition $I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1}(\tau_j, \tau_{j+1}]$
1: Set $\nu = 0$
2: **while** $J(q_\nu^j, u_\nu^j) \neq \min$ **do**
3: Replace q_ν^j in the local KKT system using (2.25), obtain local BVP for u_ν^j and z_ν^j
4: Prescribe tolerance TOL_2 and initial shooting variables $s_{\nu,0}$, set $k = 0$
5: **while** $\|F(s_{\nu,k})\|_2 > TOL_2$ **do**
6: Solve local IVP corresponding to (2.17a*)–(2.17b*), evaluate $J(q_\nu^j, u_\nu^j)$ and residual $-F(s_{\nu,k})$
7: Solve variational initial value problems (2.27), evaluate $F_s'(s_{\nu,k})$
8: Solve shooting system $F_s'(s_{\nu,k})\delta s_{\nu,k} = -F(s_{\nu,k})$
9: Compute update $s_{\nu,k+1} = s_{\nu,k} + \delta s_{\nu,k}$, set $k \leftarrow k + 1$
10: **end while**
11: Compute update $q_{\nu+1}^j$ using z_ν^j from the local IVP via (2.25)
12: **end while**

Remark 2.9. Due to the small size of the ODE side condition in (2.20) (in such context, even $100 - 500$ ODE are considered as a small system) we can compute the Jacobian matrix F_s' in each Newton step and use direct solvers. For PDE problems where (due to spatial discretization) one often has to deal with millions of degrees of freedom, direct solvers are prohibitive. This leads to using inexact Newton methods (for a description of Krylov-Newton methods and their application, see Chapters 4 and 5).

2.3.2 Direct approach

We now turn to the second, more common shooting approach, the direct multiple shooting (DMS) method, which was originally applied to the ODE optimal control framework by Bock and his co-workers (see, e. g., [10] –[13] and [15]). The basic ideas are only briefly resumed; detailed presentations can be found in the literature (see, e. g., Leineweber [73], on which the following overview is based, or the corresponding sections in Potschka [93]). The development of a direct simple shooting method is skipped here, because this essentially confronts the same stability issues already discussed above. We want to solve problem (2.20), but instead of setting up the KKT system (2.17a)–(2.17c) and applying a shooting technique to the (modified) state and adjoint equations (2.17a*) and (2.17b*), the local state variable u^j is now replaced by a function of an artificial initial value s^j and the control variable q^j, where the subintervals are determined by the decomposition (2.15). With $\bar{s} = (s^j)_{j=0}^M$ and $\bar{q} = (q^j)_{j=0}^{M-1}$, the reformulated problem reads as follows:

$$\min_{(\bar{s},\bar{q})} J(\bar{s},\bar{q}) = \sum_{j=0}^{M-1} J^j(q^j, u^j(s^j, q^j)) = \sum_{j=0}^{M-1} \int_{I_j} [\Phi(t, u^j(t; s^j, q^j(t))) + \Psi(q^j(t))]\,\mathrm{d}t \quad (2.28)$$

subject to the continuity conditions

$$\begin{aligned} s^0 - u_0 &\stackrel{!}{=} 0, \\ s^{j+1} - u^j(\tau_{j+1}; s^j, q^j(t)) &\stackrel{!}{=} 0, \quad (j = 0, \cdots, M-1) \end{aligned} \tag{2.29}$$

where u^j is the solution of the following IVP on subinterval I_j for $j = 0, \ldots, M-1$:

$$\begin{aligned} \dot{u}^j(t; s^j, q^j(t)) &= f(t, u^j(t; s^j, q^j(t))) + cq^j(t), \\ u^j(\tau_j, s^j) &= s^j. \end{aligned} \tag{2.30}$$

This setup corresponds to (2.17) in Section 2.2, except for the additional control variable in (2.29).

For problem (2.28)–(2.29) we obtain the Lagrange functional

$$\begin{aligned} \mathcal{L}((s^j, \lambda^j)_{j=0}^M, (q^j)_{j=0}^{M-1}) = &\sum_{j=0}^{M-1} J^j(q^j, u^j(s^j, q^j)) + (s^0 - u_0, \lambda^0) \\ &+ \sum_{j=0}^{M-1} (s^{j+1} - u^j(\tau_{j+1}; s^j, q^j(\tau_{j+1})), \lambda^{j+1}) \end{aligned} \tag{2.31}$$

where the $\lambda^j (j = 0, \cdots, M)$ are Lagrange multipliers corresponding to the constraints (2.29). Differentiating (2.31) w. r. t. its arguments yields the following optimality conditions

$$\mathcal{L}'_{\lambda^j}(\delta\lambda) = (s^j - g_1(u_0, u^j), \delta\lambda) \stackrel{!}{=} 0, \tag{2.25a}$$

$$\mathcal{L}'_{s^j}(\delta s) = (\lambda^j - g_2(u_s^{j\prime}), \delta s) \stackrel{!}{=} 0, \tag{2.25b}$$

$$\mathcal{L}'_{q^j}(\delta q) = J_q^{j\prime}(\delta q) + J_u^{j\prime}(u_q^{j\prime}(\delta q)) - (\lambda^{j+1}, u_q^{j\prime}(\tau_{j+1})(\delta q)) \stackrel{!}{=} 0, \tag{2.25c}$$

where

$$g_1(u_0, u^j) := \begin{cases} u_0 \text{ for } j = 0, \\ u^j(\tau_{j+1}; s^j, q^j(t)) \text{ for } j > 0, \end{cases} \qquad g_2(u_s^{j\prime}) := \begin{cases} u_s^{j\prime}(\tau_{j+1}) \text{ for } j < M. \\ 0 \text{ for } j = M. \end{cases}$$

The stationary points of (2.25a)–(2.25c) are solution candidates of the optimization problem (2.28)–(2.29). Note that differentiation of $\mathcal{L}$ w. r. t. s^j and q^j requires the solution of additional sensitivity problems similar to (2.27) in order to compute the quantities $u_s^{j\prime}$ and $u_q^{j\prime}$. These variational or sensitivity equations have the form

$$\dot{u}_s^{j\prime}(t) = f'_u(t, u^j(t; s^j, q^j(t)))u_s^{j\prime}(t), \qquad u_s^{j\prime}(\tau_j, s^j) = \delta s \tag{2.33}$$

for the sensitivity w. r. t. the initial values and

$$\dot{u}_q^{j\prime}(t) = f'_u(t, u^j(t; s^j, q^j(t)))u_q^{j\prime}(t) + c\delta q(t), \qquad u_q^{j\prime}(\tau_j, s^j) = 0 \tag{2.34}$$

for the sensitivity w. r. t. the control. These linear equations are obtained by differentiation of the state equation (2.30) w. r. t. s^j and q^j in directions δs and δq, respectively.

There are several ways to proceed from here; in the following, first the ideas underlying the implementation carried out for this thesis are presented. This works for the problem class (2.20) but is not designed to incorporate control or state constraints or differential algebraic equations (DAE) as side conditions. Afterwards, the outline of a widely used sequential quadratic programming (SQP) approach is sketched that can handle all the mentioned problem extensions. However, the latter is handled abstractly, because a detailed discussion is beyond the scope of this work and can be found in numerous articles and textbooks, for example Geiger & Kanzow [43], Nocedal & Wright [90] and Ulbrich & Ulbrich [109].

Remark 2.10. The presentation of an alternative approach is postponed to Chapter 5, because it is more suitable in the PDE context due to its potential for matrix-free computations (cf. Remark 2.9 above). While in the current situation, IMS and DMS might appear as different methods, we will show in the framework of PDE governed OCP that they are, in fact, very closely connected.

In the `MATLAB` implementation of DMS used for the results in Subsection 2.3.4, first the intervalwise state IVP (2.30) are solved for given initial values s^j and controls q^j, which enables the evaluation of the continuity conditions (2.29) of the reduced problem. As the aim is to solve the reduced KKT system by Newton's method, also the remaining equations (2.25b) and (2.25c) have to be evaluated, which necessitates the solution of equations (2.33) and (2.34). Furthermore, we need the Jacobian matrix of the KKT system (which is the Hessian of the Lagrange functional (2.31)). Without discussing this issue further, we state that here this matrix is explicitly assembled (requiring the solution of additional variational equations of type (2.34) for a whole basis of the discrete control space). This corresponds to the sensitivity approach discussed in Section 3.3, which cannot be employed in the PDE case where large-scale OCP occur. Even for large ODE problems, adjoint methods have been employed recently to increase the efficiency of multiple shooting (see, e. g., the theses of Albersmeyer [1], Beigel [8] and Potschka [94]). This alternative is pursued further in Chapter 5. Furthermore, we solve the Newton equation for our KKT system by a direct linear solver, which is also prohibitive in the PDE framework. Alternative solution routines are discussed in Chapters 4 and 5. Our proceeding for DMS implementation in the current ODE setting can be summarized by the following Algorithm 2.3:

As already mentioned above, SQP methods constitute a widely used alternative to our Newton approach. Leineweber [73] describes in detail the software package MUSCOD where SQP variants are implemented that can handle more complex problems than (2.20). Here, only the main ideas behind SQP methods are recalled, and we refer to the previously cited literature for details. For brevity, we define $y := ((s^j)_{j=0}^{M}, (q^j)_{j=0}^{M-1})$ and $\lambda := (\lambda^j)_{j=0}^{M}$. In this notation, the system (2.29) can be written as $F(y) \stackrel{!}{=} 0$, and the derivatives of (2.31) w. r. t. y and λ yield the following abstract formulation of the KKT system (2.25a)–(2.25c):

$$\begin{aligned} \nabla_y \mathcal{L}(y, \lambda) &\stackrel{!}{=} 0, \\ F(y) &\stackrel{!}{=} 0. \end{aligned} \tag{2.35}$$

The mentioned SQP approaches reduce the original problem to a sequence of approximating quadratic programming (QP) problems. In fact, one can show that the solution of one QP

Algorithm 2.3 Direct multiple shooting for ODE governed OCP

Require: Initial control $q_0 = ((q_0^j)_{j=0}^{M-1})$, initial shooting values $s_0 = ((s_0^j)_{j=0}^{M})$, decomposition $I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1}(\tau_j, \tau_{j+1}]$

1: Set $\nu = 0$
2: **while** $J(q_\nu^j, u^j(s_\nu^j, q_\nu^j)) \neq \min$ **do**
3: Solve the IVP (2.30) on the I_j
4: Solve the variational equations (2.33) and (2.34) and evaluate $\nabla\mathcal{L}$
5: Compute additional sensitivities as solutions of further variational equations and assemble the Hessian matrix $\nabla^2\mathcal{L}$
6: Solve the Newton equation for the KKT system (2.25a)–(2.25c)
7: Compute updates $s_{\nu+1}^j = s_\nu^j + \Delta s, \lambda_{\nu+1}^j = \lambda_\nu^j + \Delta\lambda$ and $q_{\nu+1}^j = q_\nu^j + \Delta q$, and set $\nu \leftarrow \nu + 1$
8: Evaluate $J(q_{\nu+1}^j, u^j(s_{\nu+1}^j, q_{\nu+1}^j))$
9: **end while**

corresponds to one step of our Newton iteration. Within these QP, one starts from a point y_0 and computes a sequence of iterates

$$y_{k+1} = y_k + \alpha_k p_k. \tag{2.36}$$

In each iteration step, a descent direction p_k and a steplength α_k are searched. The latter is usually determined by a line search or trust region algorithm. We focus on the search direction p_k, which is the solution of an appropriate quadratic approximation of (2.35) in a neighborhood of the current iterate y_k. With the abbreviation $\mathcal{L}_k := \mathcal{L}(y_k, \lambda_k)$, the quadratic approximation $\mathcal{Q}$ is obtained by a Taylor expansion of $\mathcal{L}$ around y_k:

$$\mathcal{Q}(y_k + p, \lambda_k) = \mathcal{L}_k + \nabla_y\mathcal{L}_k p + \frac{1}{2}p^\top\nabla_y^2\mathcal{L}_k p. \tag{2.37}$$

In each step of the SQP algorithm, we minimize such a quadratic problem w. r. t. the direction p and subject to a linearization

$$F_k + \nabla F_k p \tag{2.38}$$

of the original side condition (2.29). For this linear-quadratic optimization problem, the existence of a minimizer can be shown, and due to the convexity of the problem, it is sufficient to consider first order optimality conditions (see Section 3.2 for details). The minimum is denoted by p_k, and after applying one of the mentioned globalization strategies the updating step (2.36) can be performed. Having obtained an update y_{k+1}, still an update λ_{k+1} for the Lagrange multiplier is required. In the course of solving the quadratic problem (2.37), we need a different Lagrange multiplier $\tilde{\lambda}$ for the linearized side condition (2.38). The latter is determined by $\tilde{\lambda} = \lambda_{k+1} - \lambda_k$, from which an update for λ_{k+1} is directly obtained.

2.3.3 Computational aspects

In the following, techniques are discussed that contribute to the high efficiency of the DMS algorithm for ODE optimal control problems, namely the topics of control parameterization, condensing techniques for the Newton or SQP system, and sensitivity generation.

Control parameterization. In the reformulation (2.28)–(2.30) of the original OCP underlying the DMS Algorithm 2.3, the local control variable $q^j(t)$ is a function of time t. However, in the DMS context the control is usually interpreted as a piecewise polynomial of order $p \leq 3$ on the subintervals I_j, i. e.

$$q^j \equiv q^j(q_0^j, \cdots, q_p^j) \tag{2.39}$$

(e. g., $p = 0$ in the case of bang-bang control). Here, we fit the parameters $q_0^j, \cdots, q_p^j$ on the shooting interval I_j and interpolate them instead of computing a temporally distributed control function.

Remark 2.11. Approximating the function $q^j(t)$ by a low order interpolant $q^j(q_0^j, \cdots, q_p^j)$ is not part of the discretization to be discussed later on (see Chapter 4). The parameterization can already be performed within the current function space setting and has to be combined with interpolation techniques on the discrete level.

The main advantage of control parameterization is the saving of computing time and storage. In fact, the parameterization (2.39) on the shooting interval I_j with p small is to be compared with a discretization of $q^j(t)$ using the same time grid as for u^j (which usually amounts to ≥ 100 time steps on I_j, see Chapter 4). Updating the control within the Newton or SQP algorithm applied to (2.35) requires the solution of the sensitivity equations (2.34) to obtain the derivatives $u_q^{j\prime}$. With an underlying control discretization of 100 time steps on the subinterval I_j, the sensitivity equation has to be solved 100 times within the above described SQP algorithm in order to get an update for q^j. In contrast, with a control parameterization (2.39) where $p = 2$, we only have to compute two sensitivities, i. e. the sensitivity equation has to be solved only twice. Example 2.2 illustrates the gain in efficiency of control parameterization.
However, parameterizing the control results in obtaining a suboptimal state solution (belonging to the coarse control approximation) can be optained. This implies a trade-off between efficiency and accuracy as provided in Example 2.3 below. In the PDE framework, a suitable parameterization is difficult to determine without losing structural information on $q(x, t)$, which is discussed in Subsection 4.1.3. A similar concept to control parameterization being introduced in this later subsection may easily be applied in the IMS framework.

Condensing techniques. We now turn our attention to so-called condensing techniques for multiple shooting methods which aim at reducing the size of Newton's system for updating the shooting variables. They are given by s in the BVP case, by s and λ in the IMS approach for OCP, and by s, q and λ in the DMS method. In the optimal control context, they are frequently employed in combination with parameterized controls; we will see that they are less efficient with fully discretized $q^j(t)$. To clarify this conjecture, the

BVP case is briefly discussed. For the extension to OCP, we refer to the relevant literature. Recalling the definition of the block matrices $G^j(\tau_{j+1}; s^j), A$ and B from Section 2.2, Newton's equation

$$F'(\bar{s}_k)\delta\bar{s}_k = -F(\bar{s}_k) \tag{2.40}$$

with $F'(\bar{s})$ as in (2.6) and $F(\bar{s})$ denoting the system (2.17) may be written explicitly as

$$\begin{aligned} G^0(\tau_1)\delta s^0 - \delta s^1 &= -F^0, \\ G^1(\tau_2)\delta s^1 - \delta s^2 &= -F^1, \\ &\vdots \\ G^{M-2}(\tau_{M-1})\delta s^{M-2} - \delta s^{M-1} &= -F^{M-2}, \\ A\delta s^0 + B\delta s^{M-1} &= -F^{M-1}. \end{aligned} \tag{2.41}$$

Now, these equations can be rearranged in the following manner: Solve the first equation for δs^1 and insert the resulting expression into the second equation, then solve the second equation for δs^2 and insert the result into the third one and so on. This results in the system

$$\begin{aligned} \delta s^1 &= G^0(\tau_1)\delta s^0 + F^0, \\ \delta s^2 &= G^1(\tau_2)\delta s^1 + F^1 = G^1(\tau_2)[G^0(\tau_1)\delta s^0 + F^0] + F^1, \\ &\vdots \\ \delta s^M &= \left(\prod_{k=1}^{M-1} G^{M-k-1}\right)\delta s^0 + \sum_{j=0}^{M-2}\left(\prod_{k=1}^{M-j-2} G^{M-k-1}\right)F^j. \end{aligned}$$

By inserting this into the last equation

$$A\delta s^0 + B\delta s^M = -F^M$$

of system (2.41), we obtain

$$\left[A + B\left(\prod_{k=1}^{M-1} G^{M-k-1}\right)\right]\delta s^0 = -F^M - B\left[\sum_{j=0}^{M-2}\left(\prod_{k=1}^{M-j-2} G^{M-k-1}\right)F^j\right].$$

This last system (which is of the size of the dimension of the shooting variables s^j, i. e. n) is now solved instead of (2.40) (which is of size $n(M+1)$). Afterwards, the remaining update blocks are computed by sweeping through the system (2.41) in a forward manner. A similar but more complex concept is provided by Leineweber [73] for DMS in the ODE optimal control context. This approach has later been transferred to the PDE framework in the thesis of Hesse [52]. The author reduces the Newton system of DMS to the control variables; however, if we do not employ parameterization techniques, the reduction is not efficient, because the shooting variables s^j eliminated from Newton's equation by the condensing technique constitute only a small fraction of the Newton update vector in case of distributed control.
Exemplarily, an OCP with scalar ODE side condition is considered, which is split into 4

shooting intervals each discretized by 100 time steps. The uncondensed Newton system is of size $4 \cdot 100 + 4 = 404$, whereas the condensed one is reduced to the control (i. e., a size of $4 \cdot 100 = 400$). In this case, the condensing is inefficient. However, if we assume a piecewise linear control on each shooting interval (this corresponds to $p = 2$ in the above framework), the original system is of size $4 \cdot 2 + 4 = 12$, whereas the condensed one is of size $4 \cdot 2 = 8$, which corresponds to a system reduction by one third.
This particular issue is revisited in the PDE discretization context in Chapter 4, where the mentioned condensing approach by Hesse is discussed more thoroughly.

Sensitivity generation. Multiple shooting methods for both BVP and OCP require not only to solve the actual ODE system at hand, but also further linearized ODE systems providing the sensitivities w. r. t. parameters and certain arguments.
There are two different ways to obtain these sensitivities. The first 'analytical' one is based on actually solving the variational equations. In PDE optimal control, there are several paradigms on the kind of additional equations that should be solved in order to generate first and second order derivatives efficiently; these paradigms are known in the optimal control literature as the sensitivity respectively adjoint approaches toward derivative generation and are discussed in detail in Section 3.3.
In what follows, a more 'numerical' approach toward sensitivity generation is sketched. The basic feature of the following discussion is numerical differentiation. The approximation of the derivative $u'_s(t;s)$ of the state solution $u(t;s)$ w. r. t. the (parameterized) initial value s is discussed (for simplicity, we choose a simple shooting framework). In the k-th component of u we may approximate the derivative w. r. t. the i-th component s_i of s by a difference quotient (cf. Bulirsch & Stoer [20])

$$u^{k\prime}_{s_i}(t;s) \approx \frac{u^k(t;s+\delta s \cdot e_i) - u^k(t;s)}{\delta s},$$

using the i-th unit vector e_i. However, it can be shown that if the nominal solution $u(t;s)$ is determined with accuracy ε, then the difference quotient approximation yields derivatives that are accurate only up to $\sqrt{\varepsilon}$. Furthermore, working with difference quotients makes the use of variable time step size or varying ODE solvers more difficult.
An alternative first developed by Bock [11] is based on differentiating the numerical integration scheme used for solving the nominal ODE and employing the resulting integrator for computing the derivatives. Although the details on this so-called internal numerical differentiation (IND) approach are omitted, it is noteworthy that this proceeding yields the same accuracy ε as for the nominal solution. Moreover, it enables adaptive time stepping and choice of integrators, and it is equivalent to solving the variational equations.
Over the past years an extension of IND from one-step integrators to linear multistep methods (LMM), especially backward differentiation formulae (BDF), has been derived in the context of nonstationary PDE solution via ODE methods. Similarly, adjoint schemes (again, see the theses by Albersmeyer [1] and Beigel [8]), where IND is often combined with methods of automatic differentiation, have been provided.

2.3.4 Examples and comparison

In Subsections 2.3.1 and 2.3.2 the IMS and DMS methods were presented. Several aspects make them appear unrelated, the main one being the absence of any adjoint equation in DMS. In Chapter 5 this structural distinction is thoroughly discussed, and DMS will be rewritten in a manner that reveals its close relation to IMS.
The current subsection, however, is dedicated to two numerical examples illustrating the discussed results on IMS and DMS. The first example is a linear ODE control problem which shows the difference between DMS without and with control parameterization.

Example 2.2. *Consider the following problem:*

$$\begin{aligned} \min_{(q,u)} \; J(q,u) &= \frac{1}{2}\int_0^2 [(u(t)-\bar{u}(t))^2 + (q(t)-\bar{q}(t))^2]\,\mathrm{d}t \\ s.\,t. \quad \dot{u}(t) &= u(t)+q(t)-g(t), \quad u(0)=1, \end{aligned}$$

where the data functions $\bar{u}(t), \bar{q}(t)$ and $g(t)$ are given as

$$\begin{aligned} \bar{u}(t) &= \exp(2t) - \frac{1}{2}\cos(2t) + \sin(2t) + \frac{1}{2}\cos(4), \\ \bar{q}(t) &= \sin(t) - \frac{1}{2}\cos(2t) + \frac{1}{2}\cos(4), \\ g(t) &= \sin(t) - \exp(2t). \end{aligned}$$

The exact solution is then given by $u(t) = \exp(2t)$ and $q(t) = \sin(t)$.

We employ two DMS variants to solve this problem on four equidistantly distributed shooting intervals each discretized by 250 time steps of equal length. The computation is stopped if the absolute value of the distance between two successive functional values is smaller than 10^{-5}. All occurring ODE are solved by the Crank-Nicolson method. The first DMS variant operates with a full control discretization (DMS_u), resulting in 250 values $q(t_i)$ per shooting interval. In contrast, a parameterization with piecewise linear control (DMS_p) is considered, i. e., only two control values q_0 and q_1 per shooting interval. This leads to a suboptimal solution, although the state curve shown in Figure 2.4 below does not differ largely from the one depicted in Figure 2.3 that was obtained by DMS_u. Table 2.3 shows

Table 2.3. Example 2.2: DMS_u and DMS_p; the computations are performed on 4 equidistant shooting intervals, each discretized by 250 time steps. For DMS_p, a piecewise linear control parameterization is used.

	#iter	time(s)	J_{final}	$\|F\|_2$
DMS_u	2	20.8	0.467219	$1.3{\cdot}10^{-14}$
DMS_p	2	0.56	0.467272	$8.2{\cdot}10^{-15}$

that both DMS variants result in equally good functional values; besides, they both yield

shooting residuals close to zero, indicating a valid solution of the shooting system. A large amount of computing time is saved by applying DMS_p, as only three linearized problems (sensitivities w. r. t. s, q_0 and q_1) have to be solved on each shooting interval, opposed to 251 sensitivities (one for s and one for each $q(t_i)$) in the unparameterized case.

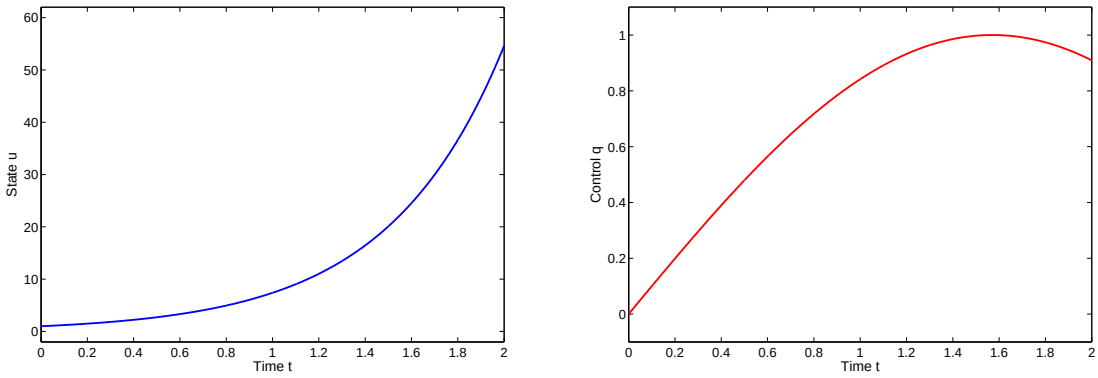

Figure 2.3. Example 2.2: DMS solution without control parameterization: state u (left) and control q (right).

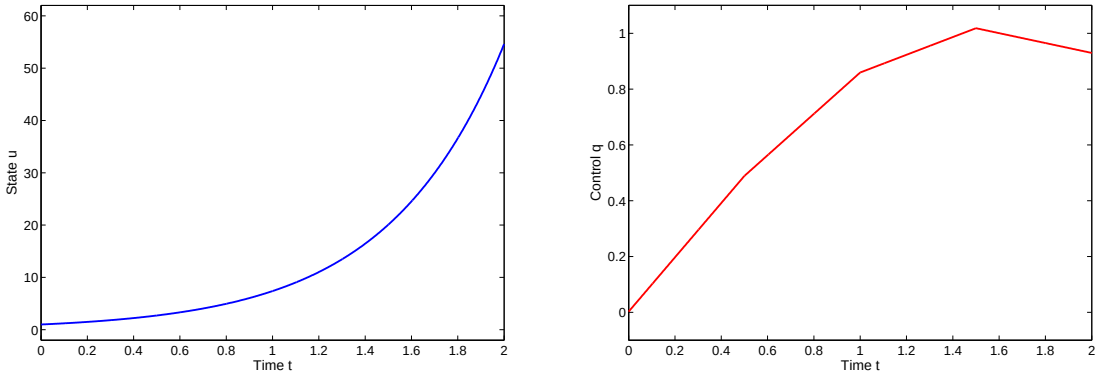

Figure 2.4. Example 2.2: DMS solution with control parameterization: state u (left) and control q (right).

Figures 2.3 and 2.4 display the state and control components after convergence. The structure of the exact solution is achieved for both cases; the state u does not seem to be affected by the control parameterization, whereas the control itself has kinks at the shooting interval transitions and is piecewise linear. The second example constitutes a sensitive optimal control problem that has been discussed in detail in Rao & Mease [97] and has been included in a benchmark collection for optimization with `MATLAB` (see Edvall & Rutquist [99]). It is suitable for illustrating the differences between IMS and DMS.

Example 2.3 (PROPT: Benchmark 51). *Consider the problem*

$$\begin{aligned} \min_{(q,u)} \; J(q,u) &= \int_0^{10} (u^2(t) + q^2(t))\,\mathrm{d}t \\ s.\,t. \quad \dot{u}(t) &= -u^3(t) + q(t), \quad u(0) = 1, \;\; u(10) = \frac{3}{2}. \end{aligned}$$

Here, the side condition is given by a nonlinear boundary value problem. The benchmark minimum value for $J(q,u)$ stated in Edvall & Rutquist [99] was computed by a collocation method and amounts to $J_{\min} = 6.723925$.

The problem is solved by the two multiple shooting approaches (IMS and DMS) on 50 equidistantly distributed shooting intervals with a time discretization of 100 timesteps per shooting interval. The solution process is stopped as soon as the computed functional value differs from the benchmark value less than $TOL = 10^{-3}$. Furthermore, the DMS method is again applied without and with piecewise linear parameterization of q on the subintervals I_j.

Table 2.4. Example 2.3: IMS and DMS; both computations are based on 100 equidistant shooting intervals each discretized by 100 timesteps. In the DMS case, a piecewise linear control parameterization is used. The reference value is $J_{\min} = 6.723925$.

	#iter	time(s)	J_{calc}	$\lvert e\rvert$	$\lVert F\rVert_2$
IMS	8	6.80	6.724074	$1.5{\cdot}10^{-4}$	$4.6{\cdot}10^{-5}$
DMS_u	5	247	6.723468	$4.6{\cdot}10^{-4}$	$7.5{\cdot}10^{-2}$
DMS_p	16	0.54	6.723061	$8.7{\cdot}10^{-4}$	$3.9{\cdot}10^{-1}$

From the data given in Table 2.4 we infer that the parameterized DMS approach is faster than IMS. In contrast, the unparameterized DMS method is comparatively slow. This is due to the huge amount of additionally solved linearized problems. IMS yields the most accurate results and fulfils the continuity conditions with higher accuracy (measured by $\|F\|_2$) than DMS. In Figures 2.5 and 2.6, the solutions of IMS and DMS are provided for the first shooting iteration and after convergence, respectively. The jumps that occur in the global solution at the beginning are distinguishable in the respective left subfigures, whereas from the right ones it is obvious that both approaches yield the same solution. In the DMS case, the adjoint component is missing; this issue is discussed in more detail in Chapter 5.

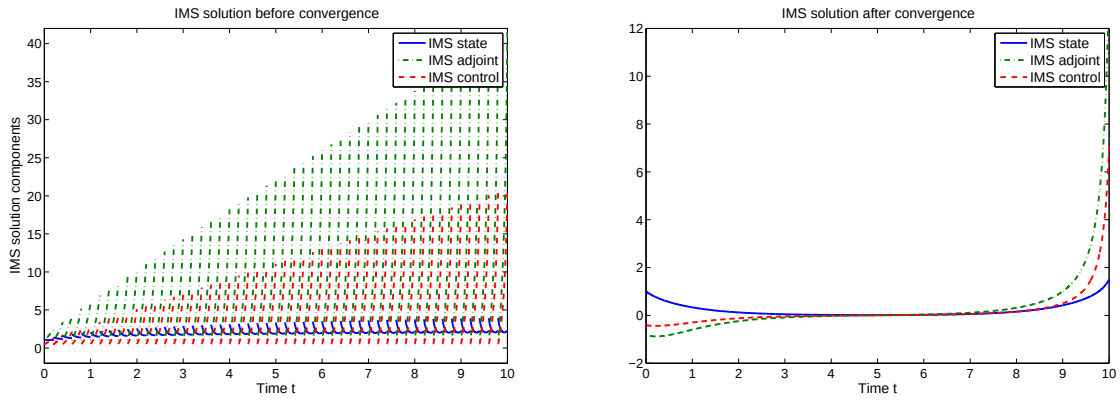

Figure 2.5. Example 2.3: IMS solution; state u, adjoint z and control q before (left) and after convergence (right).

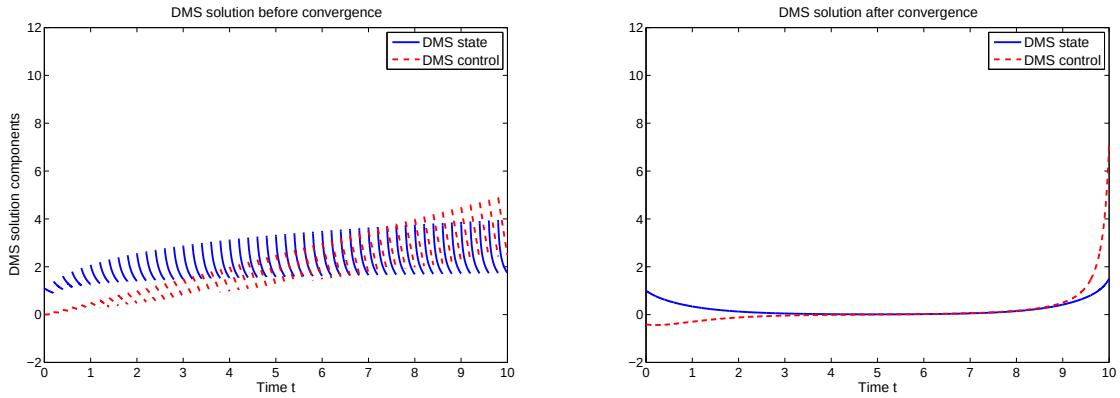

Figure 2.6. Example 2.3: DMS_u solution; state u and control q before (left) and after convergence (right).

2.4 Shooting methods for BVP in partial differential equations (PDE)

To conclude the review of multiple shooting methods for different problem classes, we present some basic parabolic PDE problems and discuss how to employ multiple shooting to solve them. However, this section merely communicates some basic impressions, e. g., obvious differences to the ODE case; the theoretical background is explained in Chapter 3, and algorithmic details are postponed to Chapter 5.
In the following, variants of the heat equation

$$\partial_t u(x,t) - \Delta u(x,t) = f(x,t) \tag{2.42}$$

are considered on a space-time cylinder $\Omega \times I$. The spatial domain is the two-dimensional unit square $[0,1]^2$, and the time interval is chosen as $[0,T]$ with $T < \infty$. A well-posed parabolic problem requires the specification of boundary values on $\partial\Omega \times I$ and an initial condition in $\Omega \times \{0\}$ at the starting time. If these demands are met, the problem is called an initial-boundary value problem (IBVP). The part $(\partial\Omega \times I) \cup (\Omega \times \{0\})$ of the boundary of the computational domain (the bottom and envelope of the space-time cylinder) constitutes the parabolic boundary. However, the notion of a PDE boundary value problem does not refer to the boundary conditions on $\partial\Omega \times I$.

Agreement. *Speaking of a PDE boundary value problem refers to boundary values on the two temporal boundaries $\Omega \times \{0\}$ and $\Omega \times \{T\}$ at the initial and final time points, respectively. If necessary for distinction, this is called a temporal parabolic BVP.*

This kind of PDE boundary value problem evokes the notion of ODE BVP discussed in Section 2.2. The different settings of an IBVP and a temporal BVP are illustrated in Figure 2.7. In fact, if such a temporal parabolic BVP is discretized by the method of lines (i. e., the spatial variables x are discretized before the time variable t), on the semidiscrete level we end up with a system of ODE which might be high-dimensional due to a fine resolution of the spatial mesh. This approach has been pursued in Potschka [94].

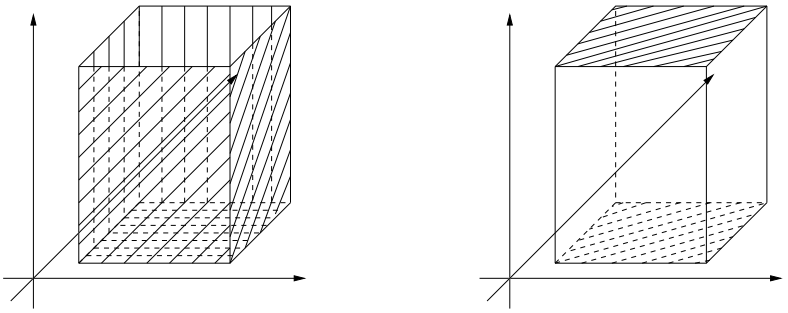

Figure 2.7. The parabolic boundary (bottom face: $\Omega \times \{0\}$, vertical axis: t)(left); the temporal boundary (bottom and top faces: $\Omega \times \{0\}$ and $\Omega \times \{T\}$)(right).

Similarly to the ODE case, we consider this type of PDE boundary value problem as it occurs in the context of parabolic optimal control as part of the system of optimality conditions. In this light, see Section 3.3, especially Remark 3.8. In the following, two concrete examples with different features are discussed, namely a time-periodic one-component problem and a two-component system with separated temporal boundary values.

A time-periodic one-component problem. The first problem to be considered is given by Example 2.4. It constitutes a single parabolic equation with periodic temporal boundary values.

Example 2.4. *Consider the following IBVP, with a periodicity condition that links the*

solution profiles at the initial and final timepoints instead of a fixed initial condition:

$$\begin{aligned} \partial_t u(x,t) - \Delta(u(x,t)) - \omega u(x,t) + \mu u(x,t)^3 &= f(x,t) && \text{in } \Omega \times I, && (2.43a)\\ u(x,t) &= 0 && \text{on } \partial\Omega \times I, && (2.43b)\\ u(x,0) &= u(x,T) && \text{in } \Omega. && (2.43c) \end{aligned}$$

The parameter $\mu \in \{0,1\}$ switches the nonlinearity on and off, and the parameter $\omega \in \mathbb{N}_0$ is responsible for a destabilization of parabolic OCP with a Helmholtz type side condition (see also Section 3.1). In the simplest case, $\mu = \omega = 0$, (2.43a) corresponds to the heat equation 2.42. Furthermore, in the computations below, we adapt the righthand side $f(x,t)$ such that the exact solution is given by

$$u(x,t) = \begin{cases} \frac{1}{4}\left(1 + \cos\left(\frac{\pi}{r_0}\|x - \tilde{x}\|_2\right)\right) & \text{if } \|x - \tilde{x}\|_2 < r_0, \\ 0 & \text{else.} \end{cases}$$

where $\tilde{x}_1 := \frac{1}{2} + \frac{1}{4}\cos(2\pi t)$ and $\tilde{x}_2 := \frac{1}{2} - \frac{1}{4}\sin(2\pi t)$.

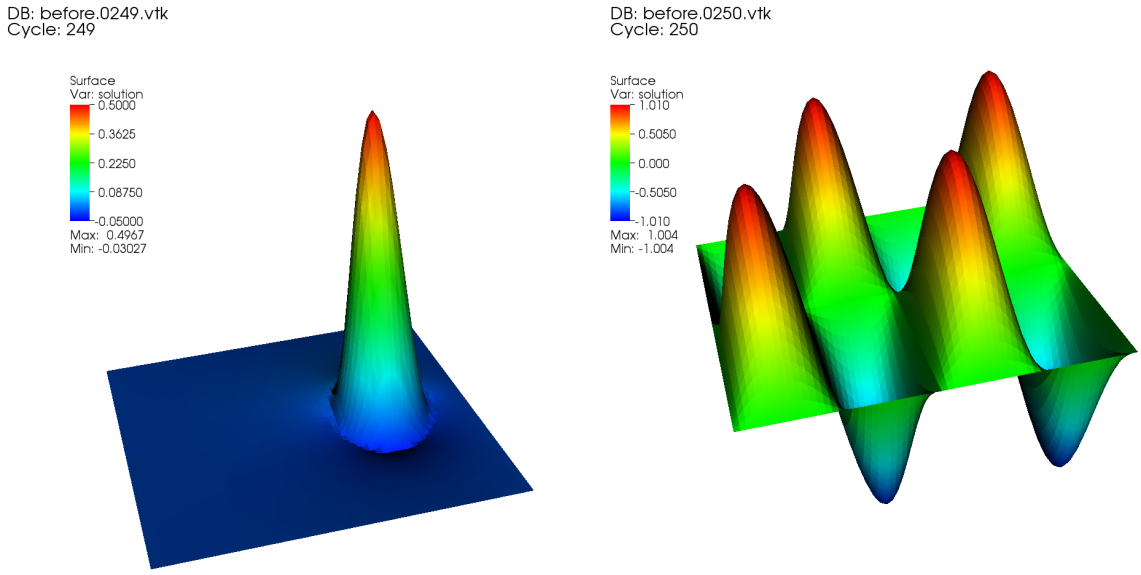

Figure 2.8. Example 2.4: Solution at $t = \frac{1}{2}$ with two consecutive timesteps before multiple shooting is converged; final timestep of first shooting interval (left); first timestep of second shooting interval (right).

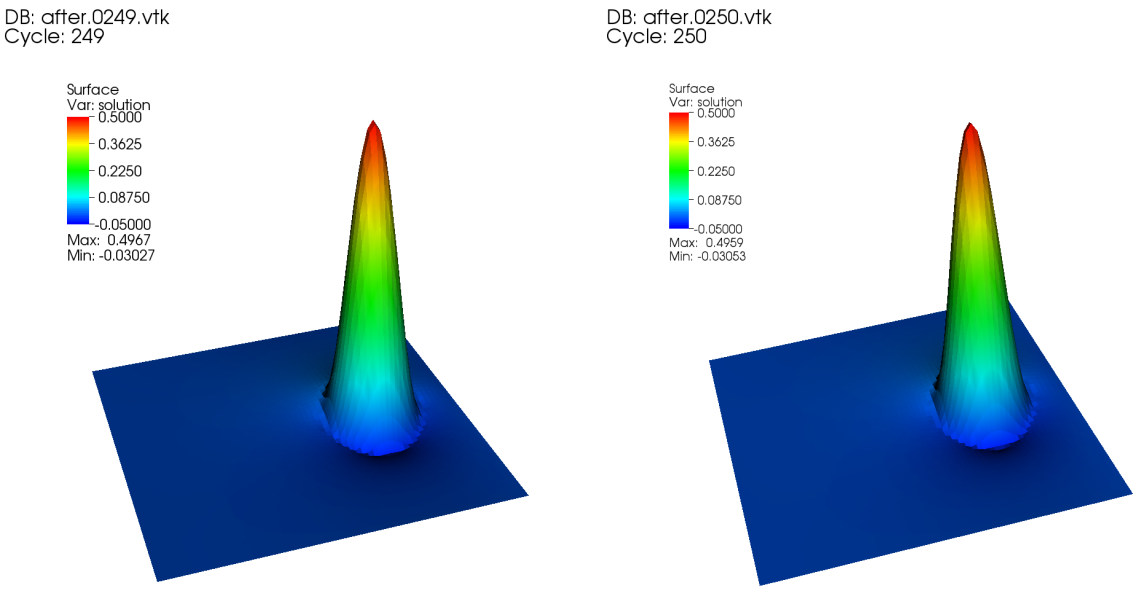

Figure 2.9. Example 2.4: Solution at $t = \frac{1}{2}$ with two consecutive timesteps after multiple shooting is converged; final timestep of first shooting interval (left); first timestep of second shooting interval (right).

On the unit square $\Omega = [0,1]^2$ and the time interval $I = [0,1]$ this function describes a bump of maximum height 0.5 which rotates on a circle with radius 0.25 around the center $(0.5, 0.5)$. The bump itself has a circular base area of radius r_0.
In order to illustrate how multiple shooting works in the PDE framework, we compute the example (2.43) with $\mu = \omega = 0$ on two shooting intervals of length 0.5, each discretized by 250 timesteps on a spatial mesh of 4096 cells. The shooting variables in the PDE case are spatially distributed functions on Ω, which we choose on each shooting interval initially as $s^{(0)}(x) = \sin(4\pi x_1)\sin(2\pi x_2)$; they describe a landscape with 4 mountains and 4 valleys on Ω and are chosen arbitrarily. Figure 2.8 displays the jump at the transition between the two shooting intervals; from the bump in timestep 249 shown in the left panel (i. e., at the end of the first shooting interval), the solution jumps to the initial value function $s^{(0)}(x)$ in timestep 250 at the beginning of the second shooting interval. A presentation of the complete solution over time, in analogy to the examples of Sections 2.2 and 2.3, is possible e. g. by creating a movie out of a sequence of timestep solutions. Alternatively, one could consider the temporal development of one fixed point in the spatial domain. Examples for this are provided in Chapter 5. In Figure 2.9, the situation after multiple shooting convergence is depicted; the jump vanishes, and the update $s^{(1)}(x)$ of the shooting variable coincides with the exact solution.
The results for different choices of μ and ω presented in Table 2.4 underline that one should use as few shooting intervals as possible. This finding corresponds to the previously discussed ODE context. In Example 2.4, one shooting interval is sufficient to solve the problem, i. e., simple shooting is stable. Although we can work with finer decompositions of

the solution interval, the table shows that an increase in the number of shooting intervals leads to a larger amount of iterations of the inexact Newton solver (here we use GMRES, see Chapter 4). Furthermore, the increase in computing time to reach the same accuracy is inacceptable.

Table 2.5. Example 2.4: Three different cases: the heat equation, the Helmholtz equation and a nonlinear equation (from left to right).

$\#_{\text{SI}}$	$\mu = 0,\ \omega = 0$			$\mu = 0,\ \omega = 7$			$\mu = 1,\ \omega = 0$		
	$\#_{\text{it}}$	$\|F\|$	time(s)	$\#_{\text{it}}$	$\|F\|$	time(s)	$\#_{\text{it}}$	$\|F\|$	time(s)
1	2	$3.0{\cdot}10^{-11}$	46	2	$1.4{\cdot}10^{-10}$	46	2	$3.0{\cdot}10^{-11}$	61
2	4	$1.7{\cdot}10^{-10}$	52	5	$3.6{\cdot}10^{-10}$	57	4	$2.4{\cdot}10^{-09}$	68
4	9	$7.9{\cdot}10^{-10}$	70	12	$1.1{\cdot}10^{-09}$	81	12	$6.0{\cdot}10^{-07}$	96
5	20	$1.2{\cdot}10^{-09}$	109	20	$1.8{\cdot}10^{-09}$	108	20	$1.8{\cdot}10^{-06}$	125
10	87	$8.7{\cdot}10^{-12}$	371	104	$1.3{\cdot}10^{-11}$	414	118	$1.8{\cdot}10^{-05}$	475

A two-component system with separated temporal boundary values. This kind of problem acts as a blueprint for later optimal control problems, where the two equations constitute the state and adjoint problems, respectively.

Example 2.5. *The second problem resembles the BVP presented in Section 2.2 in the ODE context; we are confronted with a two-component system where the initial values of the first component are imposed at $t = 0$; those of the other component are prescribed at $t = T$. The concrete problem reads:*

$$\begin{aligned}
\partial_t u_1(x,t) - \Delta(u_1(x,t)) + u_1(x,t)^2 &= f(x,t) && \text{in } \Omega \times I, && \text{(2.44a)}\\
u_1(x,t) &= 0 && \text{on } \partial\Omega \times I, && \text{(2.44b)}\\
u_1(x,0) &= u_1^0(x) && \text{in } \Omega, && \text{(2.44c)}\\
-\partial_t u_2(x,t) - \Delta(u_2(x,t)) + 2u_1(x,t)u_2(x,t) &= g(x,t) && \text{in } \Omega \times I, && \text{(2.44d)}\\
u_2(x,t) &= 0 && \text{on } \partial\Omega \times I, && \text{(2.44e)}\\
u_2(x,T) &= u_2^T(x) && \text{in } \Omega. && \text{(2.44f)}
\end{aligned}$$

Systems of this kind generalize the ODE boundary value problem (2.1) to the PDE case, where the ODE boundary values $u(a)$ and $u(b)$ are replaced by the functions $u_1^0(x)$ and $u_2^T(x)$, respectively. However, the system (2.44) does not reflect the most general case. It is not fully coupled, as the first equation does not depend on u_2. Furthermore, the boundary conditions are separated, i. e. each solution variable (u_1 resp. u_2) is fixed at exactly one temporal boundary. Suchlike problems constitute the core of the optimal control problems discussed in the remainder of this thesis, compare equations (3.32a)–(3.32b) or (3.48a)–(3.48b). Important features of such systems are the independence of the first equation on the second solution variable u_2 and the linearity as well as the backward-in-time structure of the second equation.
In the OCP framework, the two components are denoted by the terms state and adjoint

equation as in Section 2.3. However, there is an important difference between ODE and PDE problems bearing such a structure; in Subsection 2.3.1 we ignored the backward direction of the adjoint equation and solved it forward with a parameterized initial value (see (2.17a**)–(2.17b**)), which picks up the original idea of shooting methods to transform BVP into simpler IVP. However, this is not possible in the PDE framework. Hence, solving a parabolic problem backward in time, i. e., starting from the given final state at $t = T$ and searching a belonging initial state at $t = 0$, is a severely ill-posed inverse problem. In the book of Engl et al. [37], the authors explain the abstract background of this ill-posedness and illustrate it by concretely discussing the backward heat equation. From their reasoning it becomes clear that this difficulty cannot be circumvented, i. e., the second component (2.44d) – (2.44f) has to be actually solved backward in time. The algorithms achieving this are discussed in the more specific OCP context in Sections 5.1 and 5.2.
In Figures 2.10 and 2.11, we show the results of Example 2.44, being concretized as follows: The computational domain is again given as $\Omega \times I = [0,1]^2 \times [0,1]$; the data for the first component, $f(x,t)$ and $u_1^0(x)$, are chosen such that $u_1(x,t) = (t - t^2)\sin(4\pi x_1)\sin(2\pi x_2)$ is the exact solution (the structure is described and depicted in the context of Example 2.4). For the second component, we choose $g(x,t) \equiv 0$ and $u_2^T(x) = \sin(\pi x_1)\sin(\pi x_2)$. The solution is computed on two shooting intervals, where the shooting variables of the first component are initially given as $s^{(0)}(x) = \frac{1}{2}\min\{x_1, 1-x_1\}\min\{x_2, 1-x_2\}$. Those of the second component are given as $\lambda^{(0)}(x) = -2\min\{x_1, 1-x_1\}\min\{x_2, 1-x_2\}$. Figure 2.10 displays the first component at $t = \frac{1}{2}$ (timestep 250) before and after shooting. Figure 2.11 depicts the boundary value for $t = 1$ (timestep 500) in the same setting. The solver requires one inexact Newton iteration with 3 GMRES steps, which takes 521 seconds.

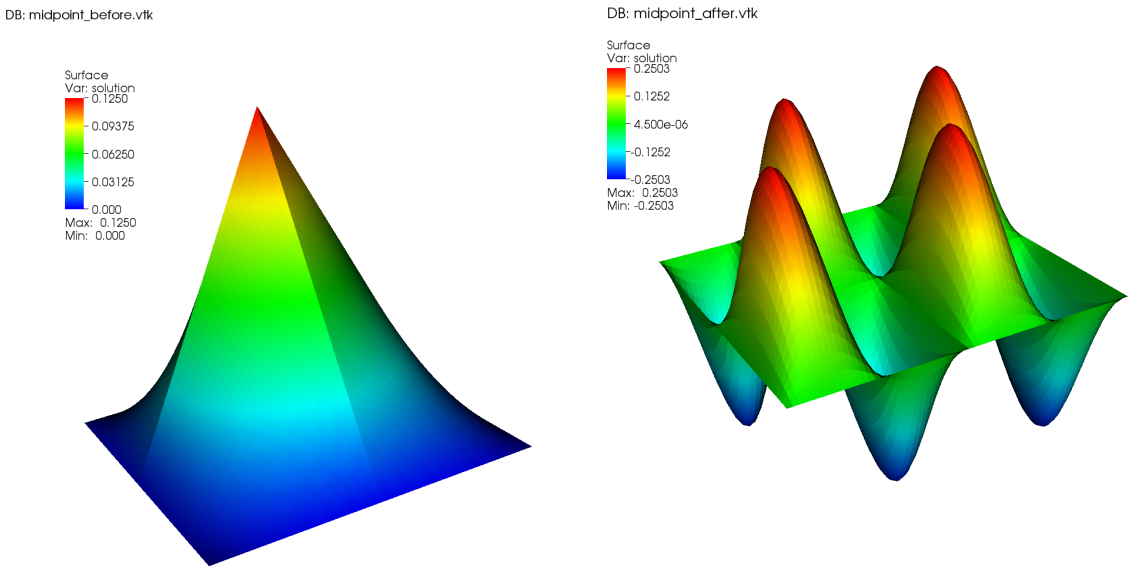

Figure 2.10. Example 2.5: Primal solution $u_1(x,t)$ at $t = \frac{1}{2}$; timestep 250 before convergence (artificial value)(left); timestep 250 after convergence (correct value up to tolerance)(right).

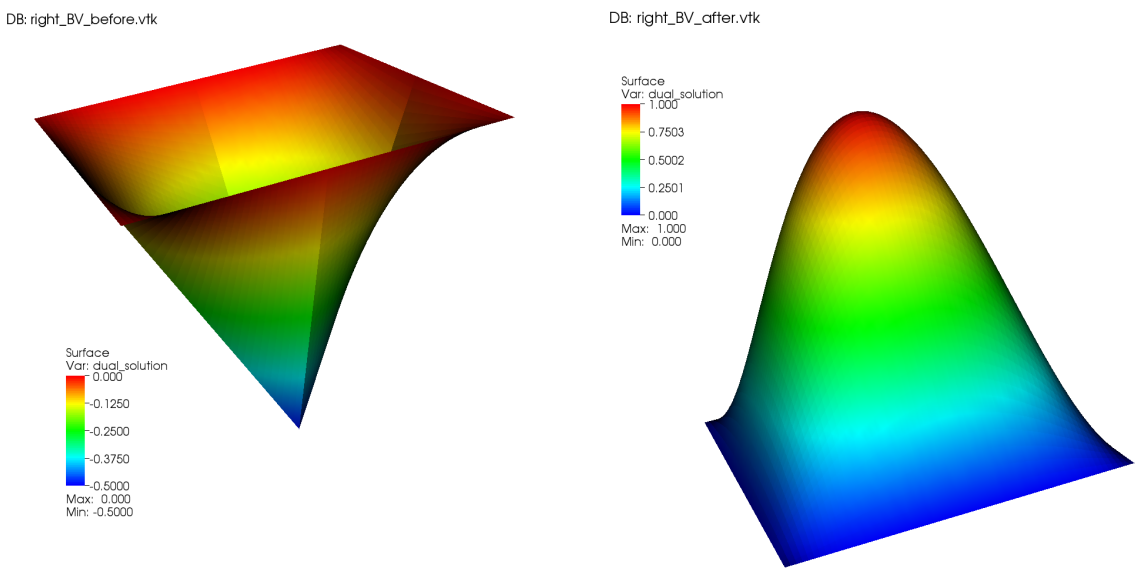

Figure 2.11. Example 2.5: Dual solution $u_2(x,t)$ at $t = 1$; timestep 500 before convergence (artificial value)(left); timestep 500 after convergence (correct value up to tolerance)(right).

3 Optimal Control Theory

This thesis focusses on optimal control problems (OCP) subject to parabolic PDE side conditions and in particular on their numerical solution by means of multiple shooting methods. The present chapter is devoted to a brief presentation of the general OCP. We inroduce the notation for an abstract OCP governed by parabolic side conditions in Section 3.1 and present examples that are discussed throughout the rest of the thesis in a modular manner. General results on existence and uniqueness for parabolic OCP are provided and applied to an example in Section 3.2. Section 3.3 is concerned with well-known optimality conditions on which our later algorithms are based; as they involve first and second order derivatives of certain operators, the optimality conditions are complemented by a presentation of ways to generate derivative information. In the final Section 3.4, we introduce non-standard modifications of the previously discussed OCP that are necessary for an embedding into the multiple shooting context. Under certain assumptions, we show the equivalence to the original control problems. Additional control constraints are covered by the theory in Section 3.2 but will be first considered in Chapter 6.

3.1 An abstract optimal control problem (OCP)

A general OCP consists in the minimization of a given functional,

$$\min_{(q,u)} J(q,u), \tag{3.1}$$

subject to a differential equation

$$e(q,u) = 0. \tag{3.2}$$

The minimum is sought in a set of functions u that fulfil a given side condition which depends on a control quantity q. The special case of an ODE side condition,

$$\dot{u}(t) = f(t,u(t),q(t)), \;\; u(t_0) = u_0,$$

on a solution interval $I = [t_0, t_f]$ has been discussed in Section 2.3. Here, $u : I \to D \subset \mathbb{R}^n$ and $q : I \to \hat{D} \subset \mathbb{R}^p$, and if $f \in C(I \times D \times \hat{D})$, which is a common regularity requirement that guarantees the existence of a solution (Peano existence theorem), then the classical spaces of continuously differentiable functions up to a certain order provide a sufficient background.

Remark 3.1. In the literature, there exists a notational inconsistency concerning the state and control variables. In the optimal control community, the control is denoted by u, the state by y, and the adjoint by p or λ. In numerical analysis of PDE, in contrast, one denotes the control by q and uses u and z for the state and the adjoint, respectively. As provided in Chapter 2, the numerical analysis nomenclature is supported throughout this thesis.

If (3.2) denotes a PDE, solving the optimization problem becomes more complicated. Depending on the given differential operator, suitable regularity assumptions have to be identified. The parabolic OCP that will dominate the remainder of this thesis reads:

$$\min_{(q,u)} \frac{\kappa_1}{2} J_1(u) + \frac{\kappa_2}{2} J_2(u(T)) + \frac{\alpha}{2} \|q\|_Q^2 \tag{3.3}$$

subject to the initial boundary value problem

$$\begin{aligned} \partial_t u(x,t) + \mathcal{A}(u(x,t)) + \mathcal{B}(q(x,t)) &= f(x,t), \\ u(x,0) &= u_0(x). \end{aligned} \tag{3.4}$$

The computational domain of the problem is a space-time cylinder $\Omega \times I$, where $\Omega \subset \mathbb{R}^d$ (for $d \in \{1,2\}$) is a bounded convex polygonal spatial domain (i. e. Ω has a Lipschitz boundary Γ) and $I = (0,T]$ is a finite time interval.
Next, we introduce the abstract function spaces suitable for problem (3.3)–(3.4). Let V and H be real Hilbert spaces of functions on Ω with a continuous and dense embedding $i : V \hookrightarrow H$. H may be identified with its dual space H^* via the Riesz representation theorem, which provides, together with the dual space V^* of V, a Gelfand triple $V \hookrightarrow H \cong H^* \hookrightarrow V^*$ on Ω. These function spaces enable an accurate description of the state $u(t)$ at a fixed timepoint t. The duality product in $V^* \times V$ is denoted by $\langle \cdot, \cdot \rangle_{V^* \times V}$. The mentioned embedding $i : V \hookrightarrow H$ and its adjoint, the embedding $i^* : H^* \hookrightarrow V^*$, permits the identification $\langle i^*(h), v \rangle_{V^* \times V} = (h, i(v))_H$ for $v \in V, h \in H$. In this regard, we may consider the duality product as a continuous continuation of the scalar product $(\cdot, \cdot)_H$. This interpretation is explained in detail by Lions [75] or Wloka [115]. We assume further that the control $q(t)$ at a fixed timepoint t lies in a Banach space R.
The full function spaces (including the time dependence) for state and control variables are usually Bochner spaces of the type $W(I;Y)$ where the time variable t is mapped into a Banach (or Hilbert) space Y. The natural setting for the parabolic PDE is the following: For given $q(x,t) \in Q := L^2(I;R)$ and righthand side $f(x,t) \in L^2(I;V^*)$, find a state function $u(x,t)$ that satisfies (3.4) and obeys additionally imposed suitable boundary conditions. Under these structural assumptions, the solution space for $u(x,t)$,

$$X := \{v(x,t) \in L^2(I;V) \,|\, \partial_t v(x,t) \in L^2(I;V^*)\}, \tag{3.5}$$

is known to be continuously embedded into the space $C(\overline{I};H)$ of temporally continuous functions with values in H (see, e. g., Dautray & Lions [30]). Thus, an initial condition $u_0(x) \in H$ is well-defined.

Remark 3.2. More generally, one may take $\tilde{X}$ as the space of functions $v(x,t) \in L^p(I;V)$ with time derivative $\partial_t v(x,t) \in L^{p'}(I;V^*)$, where p and p' are conjugate indices, i. e. $\frac{1}{p} + \frac{1}{p'} = 1$. The embedding $\tilde{X} \hookrightarrow C(\bar{I};H)$ still holds in this case.

The most important constituents of the PDE side condition are the operators. The partial derivative w. r. t. time, $\partial_t : L^2(I;V) \to L^2(I;V^*)$, is linear. The differential operator $\mathcal{A} : X \to L^2(I;V^*)$ acting on the state $u(x,t)$ is unrestricted, whereas the operator $\mathcal{B} : Q \to L^2(I;V^*)$ acting on the control $q(x,t)$ is always linear in our examples. If $R \hookrightarrow V^*$, one assumes B to be an injection operator.
In the discrete setting introduced in Section 4.1, we employ projection type methods (more precisely, Galerkin finite element methods). As the strong form of the PDE given in (3.4) is not appropriate in this context, a weak or variational formulation is derived. Therefore, need some preparatory definitions are required. We consider $\bar{\mathcal{A}} : V \to V^*$ and $\bar{\mathcal{B}} : R \to V^*$ as pointwise-in-time operators corresponding to $\mathcal{A}$ and $\mathcal{B}$, respectively, and assume that the elliptic operator $\bar{\mathcal{A}}$ is coercive. Then the following scalar products and semilinear forms can be defined:

$$(\!(u,\varphi)\!)_I := \int_I (u(t),\phi)_H \,\mathrm{d}t, \qquad a_I(u)(\varphi) := \int_I \langle \bar{\mathcal{A}}(u(t)),\phi\rangle_{V^*\times V}\,\mathrm{d}t,$$
$$b_I(q)(\varphi) := \int_I \langle \bar{\mathcal{B}}(q(t)),\phi\rangle_{V^*\times V}\,\mathrm{d}t.$$

Here, the test functions $\phi \in V$, corresponding to the pointwise-in-time operators, are distiguished from the temporally distributed test functions $\varphi \in X$. The index I denotes the integration interval (which may later on be a shooting interval or a discrete time step interval). The explicit indexing is omitted if it is evident from the context. After these notational preparations, the weak formulation of (3.4) reads: Find $u \in X$, such that for all $\varphi \in X$

$$(\!(\partial_t u,\varphi)\!) + a(u)(\varphi) + b(q)(\varphi) + (u(0),\varphi(0)) = (\!(f,\varphi)\!) + (u_0,\varphi(0)), \tag{3.6}$$

where the initial condition is weakly included.
To complete the discussion of the parabolic OCP, we finally explain the details of the objective (or cost) functional $J(q,u)$. As provided by (3.3), it comprises three terms: $J_1(u)$ is assumed to be of tracking type, i. e. $\int_I \|u(t) - \hat{u}(t)\|^2 \,\mathrm{d}t$ (where $\hat{u} \in X$ is a given function to be matched). The end-time term $J_2(u(T)) := \|u(T) - \hat{u}_T\|^2$ similarly aims at matching a given function $\hat{u}_T \in V$. Finally, the term $\frac{\alpha}{2}\|q\|_Q^2$ serves as a regularization term, where $\alpha \geq 0$ is the usual regularization parameter. In the optimal control context, it is also often regarded as measuring the costs of the control q.

Remark 3.3. In the multiple shooting framework laid out in Section 3.4, it is important that the distributed part $J_1(u)$ of the objective functional can be localized to contributions from the subintervals of a decomposition of I. Our tracking type structure ensures this property.

By imposing the conditions $\kappa_i \in \{0,1\}, \kappa_1 \neq \kappa_2$,usually either the distributed or the end-time matching term is cancelled out. In Section 2.3, for instance, we exclusively treated

distributed functionals; in the ODE context the different functional types are known as Lagrange (distributed), Mayer (end-time) or Bolza (both distributed and end-time, excluded by our conditions on κ_i) terms and can be transformed into one another. In later chapters, theoretical statements will refer to the case $\kappa_1 \neq 0$, with the required modifications for end-time matching functionals being considered in remarks.
Concluding this section, the concrete control problems that serve as examples in the later Chapters 5 – 7 are presented in a modular way. In order to test the multiple shooting method and its adaptations, simple examples suitable for highlighting certain characteristics are considered. The three main modules are the computational domain $\Omega \subset \mathbb{R}^2$, the differential operator $\mathcal{A}$ acting on the state variable u, and the objective functional $J(q, u)$. We discuss these three modules separately and list their concretizations. Certain headwords referring to the respective configuration are displayed in boldface.

The computational domain Ω. The following simple subsets of $\mathbb{R}^2$ are used as spatial domains. They are illustrated in Figure 3.1.

a) the **square** $[-1, 1] \times [-1, 1]$,

b) the **rectangle** $[-1, 3] \times [-1, 1]$.

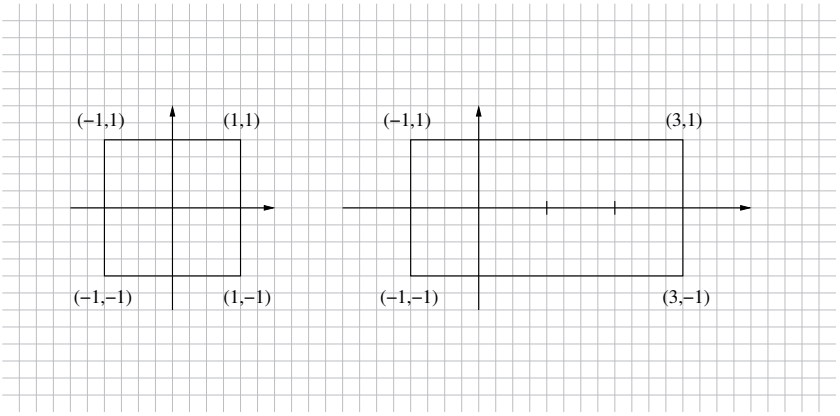

Figure 3.1. The computational domains for PDE examples in Chapters 5 – 7: the square (left) and the rectangle (right) .

The parabolic side condition (in particular, the differential operator $\mathcal{A}$). All parabolic side conditions for the OCP can be viewed as the heat equation, complemented by additional terms. These modifications often lead to a different behavior of the solution $u(x, t)$. We describe the most important modifications, discussing the respective appropriate framework (function spaces, conditions for existence and uniqueness of solutions, etc.). As starting point, the general parabolic problem is stated:

$$\begin{aligned} \partial_t u(x,t) + \mathcal{A}(u(x,t)) &= f(x,t) \quad \text{in } \Omega \times I, \\ \beta_1 u(x,t) + \beta_2 \partial_n u(x,t) &= g(x,t) \quad \text{on } \Gamma \times I, \\ u(x,0) &= u_0(x) \quad \text{in } \Omega. \end{aligned}$$

In the configurations below, the control enters the parabolic PDE always via the righthand side, where $q(x,t)$ replaces $f(x,t)$ in $\Omega \times I$. This kind of problem is denoted as distributed control problem. Linear control terms are considered exclusively. In detail, the following problems are focussed:

a) The **heat equation**:

$$\partial_t u(x,t) - \Delta u(x,t) = f(x,t) \quad \text{in } \Omega \times I. \tag{3.7}$$

Here, $\mathcal{A}(u(x,t)) = -\Delta u(x,t)$ is the Laplace operator. In case of homogeneous Dirichlet boundary values, $V = H_0^1(\Omega)$ (i. e., the boundary condition is built into the function space), $H = L^2(\Omega)$ and $V^* = H^{-1}(\Omega)$ (the dual space of $H_0^1(\Omega)$). It can then be shown that, for $f \in L^2(I; H^{-1}(\Omega))$ and $u_0 \in L^2(\Omega)$, the heat equation (3.7) has a unique solution in the space X from (3.5) (see, e. g., Wloka [115] or, for a slightly different configuration, Evans [39]). These issues are revisited in Section 3.2.

b) The **nonstationary Helmholtz equation**:

$$\partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) = f(x,t) \quad \text{in } \Omega \times I. \tag{3.8}$$

This linear equation can be regarded as part of a system of reaction-diffusion equations. It is based on the stationary Helmholtz equation $-\Delta u(x,t) = \omega u(x,t)$ with $\omega \geq 0$ which is, on the one hand, important for examining the wave equation (and other nonstationary PDE) by the method of separation of variables; the wave equation $\partial_{tt} u(x,t) - \Delta u(x,t) = f(x,t)$ can be split into a (stationary) Helmholtz equation and a second order ODE; this is not pursued any further. On the other hand, the Helmholtz equation describes the eigenvalue problem of the Laplace operator, which will become important in Chapter 5.
The same configuration is the same as for the heat equation, i. e., $V = H_0^1(\Omega)$ in the case of distributed control, together with the respectively suitable spaces H and V^* as defined above. Unique solvability is ensured by the results in Section 3.2.

c) The **nonstationary Helmholtz equation with nonlinearity**:

$$\partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + h(u(x,t)) = f(x,t) \quad \text{in } \Omega \times I. \tag{3.9}$$

The operator $\mathcal{A}(u(x,t))$ from the nonstationary Helmholtz equation is now complemented by a nonlinear term $h(u(x,t))$, i. e. $\mathcal{A}(u) = -\Delta u - \omega u + h(u)$. For $\omega < 0$, the result can be regarded as part of a system of nonlinear reaction-diffusion equations. The nonlinearity $h(u)$ has to fulfil additional conditions in order to guarantee solvability (uniqueness is not to be expected in the nonlinear case); we consider polynomial nonlinearities, e. g., $h(u) = u^3$. The space configuration can again be chosen identically to the former two examples, which will be discussed in detail below (see the end of Section 3.2).

Depending on the sign of the parameter ω, these equations can be viewed as simple prototypes for general systems of either Helmholtz type equations or reaction-diffusion equations which appear in chemical applications. The FitzHugh-Nagumo equations for

modelling the dynamics of activation and deactivation of neurons, especially the impulse conduction in the axon, are another generalization of the third example.

The objective functional $J(q,u)$. In addition to the choice between a distributed and an end-time tracking term, the functional may involve only a part $\tilde{\Omega} \subset \Omega$ of the domain or of the boundary, $\tilde{\Gamma} \subset \Gamma$. The regularization term $\frac{\alpha}{2}\|q\|_Q^2$ is influenced by the choice of the regularization parameter α. Thus, the following configurations are obtained:

a) The **distributed tracking functional** ($\kappa_1 = 1, \kappa_2 = 0$)

$$J(u,q) = \frac{1}{2} \int_I \int_\Omega \left(u(x,t) - \hat{u}(x,t)\right)^2 \, \mathrm{d}x\mathrm{d}t + \frac{\alpha}{2}\|q\|_Q^2.$$

In this case, homogeneous Dirichlet boundary conditions ($\beta_2 \equiv 0,\ g(x,t) \equiv 0$) are imposed to the PDE constraint. Thus, $V = H_0^1(\Omega)$ is appropriate, which together with $H = L^2(\Omega)$ and $V^* = H^{-1}(\Omega)$ constitutes a Gelfand triple.

b) The **end-time matching functional** ($\kappa_1 = 0, \kappa_2 = 1$)

$$J(u,q) = \frac{1}{2} \int_\Omega \left(u(x,T) - \hat{u}_T(x)\right)^2 \, \mathrm{d}x + \frac{\alpha}{2}\|q\|_Q^2.$$

Although it can be combined with arbitrary boundary conditions, this functional type will be used in the framework of distributed control and homogeneous Dirichlet boundary data.

3.2 Existence and uniqueness of solutions

This section provides a discussion of (unique) solvability of OCP such as (3.1) – (3.2). The literature yields several approaches that rely on different sets of assumptions. The most important classification concerns reduced vs. non-reduced methods. Although the reduced framework plays a prominent role in later chapters, here the non-reduced approach is focussed. This enables more general existence results that comprise all the above examples. The alternative approaches are set into relation. The proofs within this section are based on Fursikov [40].
The section comprises three parts, two abstract theorems for OCP subject to linear and nonlinear side conditions (3.2), respectively, and a concrete result that is tailored to the above mentioned nonlinear Helmholtz equation of reaction-diffusion type.

Preparations. Before starting with the linear case, the basic framework is developped. We consider the abstract extremal problem

$$\min_y \ J(y) \quad \text{subject to} \quad F(y) = 0, \ \ y \in D \subset Y. \tag{3.10}$$

Y is a normed vector space, $D \subset Y$ is a closed and convex subset (i.e., for each two $y_1, y_2 \in D$, the connecting line segment $\{y \mid y = \lambda y_1 + (1-\lambda)y_2, \lambda \in (0,1)\}$ is fully

contained in D). Furthermore, the functional $J : D \to \mathbb{R}$ is assumed as weakly lower semi-continuous, i. e.,

$$\liminf_{n\to\infty} J(y_n) \geq J(y) \qquad (y_n \rightharpoonup y),$$

(the symbol $\rightharpoonup$ denoting, as usual, weak convergence), and as bounded from below, i. e., for all $y \in D$ there is $c \in \mathbb{R}$ with

$$c \leq J(y) \leq \infty.$$

The operator $F : \hat{Y} \to Z$ from the side condition is assumed to be continuous. Here, $\hat{Y}$ is a reflexive Banach space which is continuously embedded into Y, and Z is a normed vector space.

Definition 3.1. *An element $y \in \hat{Y}$ is called admissible for problem* (3.10) *if $F(y) = 0$, $y \in D$ and $J(y) < \infty$. The set of all admissible elements is denoted by $\hat{Y}_{ad}$.*

The following two assumptions conclude the preparation.

Assumption 3.1. *(i) The admissible set $\hat{Y}_{ad} \subset \hat{Y}$ is nonempty. (ii) For each $\kappa > 0$ the set $\{y \in \hat{Y}_{ad} \mid J(y) < \kappa\}$ is bounded in $\hat{Y}$.*

Remark 3.4. The condition $y \in D$, where $D \subset Y$ is closed and convex, is sufficiently general to comprise the case of additional control or state constraints. This case is revisited in Chapter 6. In the current framework, $D = Y$ is assumed.

The linear case. The operator $F : \hat{Y} \to Z$ is given as $F(y) = Ly + F_0$ with a linear continuous operator $L : \hat{Y} \to Z$ and an element $F_0 \in Z$. Then the following theorem holds. As the proof makes use of some notions that are important in the optimization context, we present it as well. This also simplifies the verification of certain assumptions in concrete examples.

Theorem 3.1. *Let the assumptions and conditions of the preparatory subsection hold, and let $F : \hat{Y} \to Z$ be an affine linear operator. Then problem* (3.10) *has a solution $\hat{y} \in \hat{Y}$. If, in addition, the functional $J : D \to \mathbb{R}$ is strictly convex, i. e. if for all $y_1, y_2 \in D$ with $y_1 \neq y_2$ and for all $\lambda \in (0, 1)$*

$$J\left(\lambda y_1 + (1-\lambda)y_2\right) < \lambda J(y_1) + (1-\lambda)J(y_2),$$

then the solution of (3.10) *is unique.*

Proof. By Assumption 3.1 (i) $\hat{Y}_{ad}$ is nonempty, thus there exists a minimizing sequence $(y_n)_{n\in\mathbb{N}} \subset \hat{Y}_{ad}$ with $\lim_{n\to\infty} J(y_n) = \inf_{y\in\hat{Y}_{ad}} J(y)$. The convergent sequence $(J(y_n))_{n\in\mathbb{N}} \subset \mathbb{R}$ is then bounded, i. e., $J(y_n) \leq \kappa < \infty$ holds uniformly. Assumption 3.1 (ii) yields the boundedness of $(y_n)_{n\in\mathbb{N}}$ in $\hat{Y}_{ad}$. From the latter the existence of a weakly convergent subsequence $(y_{n_k})_{k\in\mathbb{N}} \subset \hat{Y}_{ad}$ with limit $\hat{y}$ is obtained. Due to the continuous embedding $\hat{Y} \hookrightarrow Y$, and recalling the definition of weak convergence, the weak convergence $y_{n_k} \rightharpoonup \hat{y}$ can be inferred also in Y. Since $D \subset Y$ is closed and convex, it is also weakly sequentially closed

which implies $\hat{y} \in D$. Using the adjoint operator $L^* : Z^* \to \hat{Y}^*$, the weak convergence $(y_{n_k})_{k\in\mathbb{N}} \subset \hat{Y}_{ad}$ entails for all $v \in Z^*$:

$$\langle v, Ly_{n_k}\rangle_{Z^*\times Z} = \langle L^*v, y_{n_k}\rangle_{\hat{Y}^*\times\hat{Y}} \to \langle L^*v, \hat{y}\rangle_{\hat{Y}^*\times\hat{Y}} = \langle v, L\hat{y}\rangle_{Z^*\times Z}.$$

Due to $(y_{n_k})_{k\in\mathbb{N}} \subset \hat{Y}_{ad}$, $Ly_{n_k} + F_0 = 0$ holds, from which $L\hat{y} + F_0 = 0$ can be inferred. By assumption, J is weakly lower semicontinuous, and therefore the weak convergence $y_{n_k} \rightharpoonup \hat{y}$ in Y yields $J(\hat{y}) \le \liminf_{k\to\infty} J(y_{n_k})$. This means $\hat{y} \in \hat{Y}_{ad}$, thus $\hat{y}$ solves problem (3.10).
Assume strict convexity of J and let y_1 and y_2 be two different solutions. As $\hat{Y}_{ad}$ is convex, the line segment between y_1 and y_2 is contained in $\hat{Y}_{ad}$, specifically the point $\frac{y_1+y_2}{2}$. Strict convexity of J yields $J(\frac{y_1+y_2}{2}) < \frac{1}{2}J(y_1) + \frac{1}{2}J(y_2)$, and thus neither y_1 nor y_2 can be a minimizer. □

This abstract result on general extremal problems may be concretized to problem (3.1) – (3.2) as follows. The abstract space Y from above is replaced by $Q \times X$ with X as in (3.5) and $Q \subset L^2(I; R)$ the control space. Note that the range space V and its dual V^* in the definition of X are chosen according to the type of prescribed boundary condition. As $Q\times X$ is a Banach space, we may assume that $\hat{Y} \equiv Y$ in our concrete configuration. Consider further a closed convex subset $Q_{ad} \subset Q$ (until Chapter 6, we take $Q_{ad} = Q$, see Remark 3.4), and let the objective functional $J(q, u)$ be defined on $\hat{Y}_{ad} \equiv Y_{ad} := Q_{ad} \times X$. Section 3.3 states that in this framework J (as defined in (3.3)) fulfils the abstract assumptions of continuity, boundedness from below, and strict convexity. Furthermore, the abstract affine linear operator $Ly + F_0$ is given by the linear parabolic PDE $e(q, u) = 0$ (e. g. the heat equation or the nonstationary Helmholtz equation) which comprises the affine part F_0 as a potential non-homogeneous righthand side f. Here, $e : Q_{ad} \times X \to L^2(I; V^*)$, where the range space is a concrete instantiation of Z. The admissible set Y_{ad} consists of all $(q, u) \in Q_{ad} \times X$ that fulfil $e(q, u) = 0$ and $J(q, u) < \infty$. By applying Assumption 3.1 in this context, we obtain:

Corollary 3.2. *Problem* (3.1) – (3.2) *has a unique solution* $(\hat{q}, \hat{u}) \in Q_{ad} \times X$.

Remark 3.5. In the introduction to this section, the reduced approach was mentioned; it interprets the state u as a function of the control q. For the sake of completeness, we state that in the reduced framework, classical existence theorems from variational calculus (see, e. g., Dacorogna [29]) can be applied. However, to obtain the relationship $u = u(q)$, unique solvability of the parabolic side condition $e(q, u(q)) = 0$ has to be guaranteed, which implies the existence of a solution operator $S : Q \to X$. In the linear framework, this is easily obtained (see, e. g., Wloka [115]), but for nonlinear examples, the uniqueness condition on the PDE solution is not satisfiable in general; due to the resulting lack of a solution operator, the reduced approach will not work in this case.

The nonlinear case. In addition to the assumptions of the preparatory subsection, a normed vector space Y^* is considered for which the following embeddings hold:

$$Y \hookrightarrow Y^* \text{ (continuous embedding)}, \quad \hat{Y} \hookrightarrow\hookrightarrow Y^* \text{ (compact embedding)}.$$

Furthermore, $F : \hat{Y} \to Z$ is assumed to be a nonlinear continuous operator which fulfils the following

Assumption 3.2. *Let there be an everywhere dense subset $S \subset Z^*$ so that, for each $v \in S$, the continuous continuation of the functional $y \mapsto \langle v, F(y)\rangle_{Z^*\times Z}$ from $\hat{Y}$ to Y^* exists.*

The extremal problem (3.10) is nonlinear in this case, but the admissible set $\hat{Y}_{ad}$ is given as in Definition 3.1. Let the Assumptions 3.1 and 3.2 and the other prerequisites be fulfilled, then one can show

Theorem 3.3. *There is a solution $\hat{y} \in \hat{Y}$ of the nonlinear problem* (3.10).

Proof. This formulation is based on Fursikov [40], where the proof is included. □

Note that Theorem 3.3 comprises no statement on the uniqueness of the solution, which is to be expected in the general nonlinear case. Furthermore, the additional conditions, especially the required embeddings, have to be verified in each concrete example.
This section is concluded by recapitulating an existence result for a distributed tracking functional subject to the nonstationary nonlinear Helmholtz equation, thereby formulating conditions on the nonlinearity.

A concrete existence result for a nonlinear problem. In the following, conditions on the term $h(u)$ from the nonlinear example above are discussed, some basic estimates are repeated, and a concrete existence result is stated. Instead of presenting the proofs (which can be found in Fursikov [40]) the results are compared to Hinze et al. [59] or Tröltzsch [108]. It can be seen from our presentation and the cited literature that even for simple semilinear parabolic equations, the theory behind is nontrivial.
Consider the problem

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) + h(u(x,t)) &= q(x,t) \quad \text{in } \Omega \times I, \\ u(x,t) &= 0 \qquad \text{on } \Gamma \times I, \\ u(x,0) &= u_0(x) \quad \text{in } \Omega. \end{aligned} \tag{3.11}$$

Later on, the following conditions are required to be fulfilled, which is the case for many standard nonlinearities such as polynomials $h(u) = \pm u^n$ ($n \in \mathbb{N}$), exponential functions $h(u) = \pm e^{\pm u}$, or the trigonometric functions $h(u) = \sin(u)$ and $h(u) = \cos(u)$.

Assumption 3.3. *(i) Let one of the following conditions hold: $\sup_{u\geq 1}\left|\frac{h(u)}{u}\right| < \infty$ or $\lim_{u\to\infty}\left|\frac{h(u)}{u}\right| = \infty$. If $\frac{h(u)}{u} \to -\infty$ for $u \to \infty$, then there are constants $C_1 > 0$, $C_2 > 0$ so that for $u > 0$ the following inequality holds:*

$$\Phi(u) := \int_0^u h(\lambda)\,\mathrm{d}\lambda \geq \left(C_1 - \frac{1}{2}\right)\left|\frac{h(u)}{u}\right| - C_2.$$

(ii) Let one of the following conditions hold: $\sup_{u\leq -1}\left|\frac{h(u)}{u}\right| < \infty$ or $\lim_{u\to-\infty}\left|\frac{h(u)}{u}\right| = \infty$. If $\frac{h(u)}{u} \to \infty$ for $u \to -\infty$, then there are constants $C_1 > 0$, $C_2 > 0$ so that for $u < 0$ the inequality for $\Phi(u)$ holds.

The assumption is checked only for the case $h(u) = u^3$ which is the standard nonlinearity in later chapters. We have $\lim_{u\to\infty} \left|\frac{u^3}{u}\right| = \lim_{u\to\infty} u^2 = \infty$ and $\lim_{u\to-\infty} \left|\frac{u^3}{u}\right| = \lim_{u\to\infty} u^2 = \infty$. Due to the second limit and according to Assumption 3.3 (ii), the inequality for Φ has to be checked in the case $u < 0$. This results in

$$\Phi(u) = \int_0^u h(\lambda)\,\mathrm{d}\lambda = -\int_u^0 \lambda^3\,\mathrm{d}\lambda = \left[-\frac{1}{4}\lambda^4\right]_u^0 = \frac{1}{4}u^4 \geq \frac{1}{4}u^2 - 1.$$

Thus, the inequality for Φ holds with $C_1 = \frac{3}{4}$ and $C_2 = 1$.
For the existence result of Theorem 3.4, the following definition of the space $X_{2,1}$ is required.

Remark 3.6. The solution space X for the parabolic problems (see (3.5)) is a Hilbert space, which is proved, e. g., by Wloka [115]. As the continuous embedding $X \hookrightarrow C(\bar{I}; H)$ holds, we can infer that X is a subspace of $L^2(I; H)$.

Definition 3.2. *We consider the subspace $X_{2,1}$ of $L^2(I; L^2(\Omega))$:*

$$X_{2,1} = \{u(x,t) \in L^2(I; L^2(\Omega)) \mid \|u\|^2_{X_{2,1}} < \infty\} \tag{3.12}$$

where the norm $\|u\|_{X_{2,1}}$ is defined as

$$\|u\|_{X_{2,1}} := \sqrt{\int_I \int_\Omega [(\partial_t u(x,t))^2 (T-t)^2 + |\nabla u(x,t)|^2 (T-t)]\,\mathrm{d}x\mathrm{d}t}.$$

The nontriviality of the admissible set postulated in Assumption 3.1 (i) reads in the current context: There is a pair $(q, u) \in Q_{ad} \times X_{2,1}$ fulfilling (3.11), so that in addition

$$\int_I \int_\Omega (T-t)|h(u)u|\,\mathrm{d}x\mathrm{d}t < \infty. \tag{3.13}$$

The central result on OCP with semilinear parabolic side conditions unfolds as:

Theorem 3.4. *There is a solution $(\hat{q}, \hat{u}) \in Q_{ad} \times X_{2,1}$ to the problem*

$$\min_{q,u} J(q,u) := \frac{1}{2}\|u - \bar{u}\|^2_X + \frac{\alpha}{2}\|q\|^2_Q \;\; s.\,t. \;\; (3.11),$$

and (3.13) *is fulfilled for $\hat{u}$.*

Proof. The proof relies on an a priori estimate for functions $u \in X_{2,1}$ that fulfil (3.13). Both the estimate and the proof can be found in Fursikov [40]. □

Remark 3.7. The important feature of the presented configuration is that $X_{2,1}$ is a subset of $L^2(I; L^2(\Omega))$. In all examples discussed in Section 3.1, the Hilbert space H is chosen as $L^2(\Omega)$, which results in $X_{2,1} \subset L^2(I; H)$. Thus, regarding Remark 3.6, the same function spaces can be used as for the linear examples if only the conditions of the current subsection

are fulfilled, which enables a unified treatment of all our examples. Sometimes, a more restrictive alternative is used where $q(x,t)$ has to lie in $L^r(I;L^r(\Omega))$ with $r > \frac{d}{2}+1$. In our two-dimensional examples ($d=2$), this would necessitate $r>2$, whereas in three spatial dimensions, r would have to be chosen even larger than $\frac{5}{2}$. A similar restriction is made also for the Neumann boundary control case; for these alternative configurations, see the textbooks of Hinze et al. [59] or Tröltzsch [108].

3.3 Optimality conditions and derivative generation

As a basis for the solution methods presented in later chapters, now necessary and sufficient optimality conditions for OCP are discussed. As most algorithms are formulated in terms of reduced optimal control, these properties are derived in the reduced framework. The optimality conditions involve derivatives of the objective functional as well as the PDE differential operators. Therefore, the most important notions concerning differentiation in function space are recalled. The last part of this section deals with methods for actually computing the first and second order derivatives that are needed for evaluating the optimality conditions.

Preparations. In the reduced approach, the set of independent variables of the OCP (given by the control q and the state u in problem (3.1) – (3.2)) is reduced to the control variable. Therefore, the existence of a solution operator $S: Q \to X$ for (3.2) is postulated that maps a given control to the corresponding state. This holds if the state u can be expressed in a unique fashion as a function of the control q, i. e. $u = u(q)$. In the following, both notations $S(q)$ and $u(q)$ will be used on an equal footing. The dependence on q is characterized by the following reformulation of the weak PDE (3.6):

$$(\!(\partial_t u(q), \varphi)\!) + a(u(q))(\varphi) + b(q)(\varphi) + (u(q)(0), \varphi(0)) = (\!(f, \varphi)\!) + (u_0, \varphi(0)). \tag{3.14}$$

The OCP (3.1) subject to (3.14) can now be expressed in the following form:

$$\min_q \hat{J}(q) := J(q, u(q)). \tag{3.15}$$

Note that this OCP is unconstrained on Q, i. e., the side condition (3.14) is regarded as solved. The functional $\hat{J}(q)$ is called the reduced objective functional. This concept can be extended to more general situations with additional control or state constraints (see Chapter 6), and in Chapter 5 a modified reduction strategy will be encountered in the context of direct multiple shooting (DMS).

The formulation of the optimality conditions for the reduced problem (3.15) involves directional derivatives, as well as differentials of the functional $\hat{J}$. Therefore, we need the following concepts from functional analysis, which can be found in Hinze et al. [59] and Tröltzsch [108]. Let U and V be real Banach spaces, $\mathcal{U}$ an open subset of U and $f: \mathcal{U} \to V$ a mapping.

Definition 3.3. *If for given $u \in \mathcal{U}$ and $h \in U$ the limit*

$$\delta f(u)(h) := \lim_{t \searrow 0} \frac{f(u+th) - f(u)}{t}$$

exists as an element of V, then it is called the directional derivative of f in u in direction h. If the limits exists for all directions $h \in U$ then f is called directionally differentiable in u and the mapping $h \mapsto \delta f(u)(h)$ is called the first variation of f in u.

This generalizes the concept of directional differentiability known from the finite dimensional Euclidean space. A mapping which is directionally differentiable in u is not necessarily continuous in u, and the first variation does not have to be a linear mapping.

Definition 3.4. *If the first variation $\delta f(u)(h)$ in u exists and coincides with a linear continuous operator $A : U \to V$, i. e. $\delta f(u)(h) = Ah$ for all $h \in U$, then f is called Gâteaux differentiable in u, and A is the Gâteaux derivative of f in u, in short $A = f'(u)$.*

The Gâteaux derivative may be computed as a directional derivative. If f is Gâteaux differentiable in u, the $f'(u)$ is an element of the dual space U^*.

Definition 3.5. *The mapping $f : \mathcal{U} \to V$ is called Fréchet differentiable in $u \in \mathcal{U}$ if there is a linear continuous operator $A : U \to V$ and a mapping $r(u, \cdot) : U \to V$ so that for all $h \in U$ with $u + h \in \mathcal{U}$, it holds*

$$f(u+h) = f(u) + Ah + r(u,h) \quad \textit{with} \quad \lim_{\|h\|_U \to 0} \frac{\|r(u,h)\|_V}{\|h\|_U} = 0.$$

The linear operator A is called the Fréchet derivative of f in u, in short $A = f'(u)$.

Fréchet differentiability of a mapping f in u implies Gâteaux differentiability of f in u; it is a generalization of the finite dimensional concept of total differentiability to Banach spaces. All differentiability concepts are local but hold on the subset $\mathcal{U} \subset U$ if they hold in every single point $u \in \mathcal{U}$. The chain rule holds for Fréchet and Gâteaux derivatives, and the concepts can be transferred to higher order derivatives.

Agreement. *Further on, the objective functional $J : Q \times X \to \mathbb{R}$ and the differential operator $e : Q \times X \to L^2(I; V^*)$ are assumed to be sufficiently regular, which means at least twice continuously Fréchet differentiable. This has to be checked anew for each specific problem.*

Optimality conditions. While the existence results of Section 3.2 have a global character (they simply confirm existence of a solution), the following results are based on a more local point of view. As stated before, one cannot always expect a unique optimum, but there may be several solutions to a given OCP, each of which is an optimum possibly only within a certain neighborhood. Only under additional restrictive assumptions, a local optimum is also known to be global. A point $\hat{q}$ is said to be a local solution of the reduced problem (3.15) if there is a neighborhood $\mathcal{D} \subset Q$ of $\hat{q}$ so that $\hat{J}(\hat{q}) \leq \hat{J}(q)$ for all $q \in \mathcal{D}$.
The following constitutes an important first order optimality condition for problem (3.15).

Theorem 3.5. *(a) Let the reduced objective functional $\hat{J}$ be Gâteaux differentiable within an open set $\mathcal{D} \subset Q$. If $\hat{q} \in \mathcal{D}$ is a local optimum of problem* (3.15)*, then it holds*

$$\hat{J}'_q(\hat{q})(\delta q) = 0 \qquad \forall\, \delta q \in Q. \tag{3.16}$$

(b) If $\hat{J}$ is convex, then each $\hat{q}$ fulfilling (3.16) *is a local minimum of $\hat{J}$.*

Proof. (a) Let $\hat{q}$ be a minimum and δq be an arbitrary direction. As $\mathcal{D}$ is open, there is $\alpha \in \mathbb{R}$ so that $\hat{q} \pm \alpha\delta q$ are both in $\mathcal{D}$. As $\hat{q}$ is an optimal solution, one obtains $\alpha^{-1}(\hat{J}(\hat{q} + \alpha\delta q) - \hat{J}(\hat{q})) \geq 0$ and $\alpha^{-1}(\hat{J}(\hat{q} - \alpha\delta q) - \hat{J}(\hat{q})) \geq 0$. Passing to the limit $\alpha \to 0$, this yields both $\hat{J}'_q(\hat{q})(\delta q) \geq 0$ and $-\hat{J}'_q(\hat{q})(\delta q) \geq 0$, from which the result immediately follows.
(b) By a standard argument, the convexity of $\hat{J}$ implies $\hat{J}(q) - \hat{J}(\hat{q}) - \hat{J}'_q(\hat{q})(\delta q) > 0$. By (3.16), this results in $\hat{J}(q) \geq \hat{J}(\hat{q})$. □

If $\mathcal{D}$ is assumed as convex and closed, the equation (3.16) turns into the variational inequality $\hat{J}'_q(\hat{q})(\hat{q} - \delta q) \geq 0$ for all $\delta q \in \mathcal{D}$. This will be discussed in Chapter 6.
The next theorem contains a second order necessary optimality condition for problem (3.15).

Theorem 3.6. *If the reduced objective functional $\hat{J}$ is twice continuously Fréchet differentiable within an open set $\mathcal{D} \subset Q$, and if $\hat{q} \in \mathcal{D}$ is a local minimum of problem* (3.15)*, then it holds*

$$\hat{J}''_{qq}(\hat{q})(\delta q, \delta q) \geq 0 \qquad \forall\, \delta q \in Q. \tag{3.17}$$

Proof. As $\mathcal{D}$ is open, there is $\alpha \in \mathbb{R}$ so that $\hat{q} + \alpha\delta q$ is in $\mathcal{D}$. By Taylor expansion, one obtains due to the optimality of $\hat{q}$

$$0 \leq \hat{J}(\hat{q} + \alpha\delta q) - \hat{J}(\hat{q}) = \alpha\hat{J}'_q(\hat{q})(\delta q) + \frac{\alpha^2}{2}\hat{J}''_{qq}(\hat{q})(\delta q, \delta q) + \mathcal{R}(\alpha\delta q),$$

where the remainder term $\mathcal{R}(\alpha\delta q)$ is of third order in $\alpha\delta q$. Due to the optimality of $\hat{q}$, the first order necessary condition holds and division by $\alpha^2/2$ yields $0 \leq \hat{J}''_{qq}(\hat{q})(\delta q, \delta q) + 2/\alpha^2\mathcal{R}(\delta q)$. When passing to the limit $\alpha \to 0$, the remainder term vanishes and the stated resultis achieved. □

Finally, a second order sufficient optimality condition is quoted. The proof relies again on Taylor expansion and is omitted (see, e. g., Hinze et al. [59] for details).

Theorem 3.7. *Let the reduced objective functional $\hat{J}$ be twice continuously Fréchet differentiable within an open neighborhood $\mathcal{D} \subset Q$ of q. Further, let q fulfil the first order necessary condition* (3.16)*. Assume there is $\gamma > 0$, such that the second order sufficient condition*

$$\hat{J}''_{qq}(q)(\delta q, \delta q) \geq \gamma\|\delta q\|_Q^2 \tag{3.18}$$

holds for all $\delta q \in \mathcal{D}$. Then there are $r > 0$ and $\sigma > 0$, so that the quadratic growth condition

$$\hat{J}(q + \delta q) \geq \hat{J}(q) + \sigma\|\delta q\|_Q^2$$

holds for all $\delta q \in \mathcal{B}_r(q) \subset \mathcal{D}$. The latter is a reformulation of the optimality of q.

A method for sensitivity computation. Common optimality conditions are based on differentiability properties of the reduced objective functional. Therefore, computable representations of the first and second order derivatives of $\hat{J}$ given by $\hat{J}'_q(q)$ and $\hat{J}''_{qq}(q)$ are required. We present a way of computing these derivatives by solving certain additional equations; the formulation is done in a function space setting, but a discrete analogue is provided in Chapter 4.
There are two main methods for derivative computation, the sensitivity approach and the adjoint approach, which are both applicable for first as well as second order derivatives. As our further discussion shows, the sensitivity approach is not efficient in the parabolic OCP context; therefore, the focus is on the adjoint method, which also underlies all practical PDE examples presented in this thesis. Besides an explicit derivation, it will also be shown how to embed the adjoint method into the Lagrange formalism context which becomes important for later developments. The presentation is mainly influenced by Hinze et al. [59] and Meidner [84], where detailed presentations of the sensitivity approach are included as well.

For convenience, the OCP under consideration is restated in an abstract form which is tailored to the following discussion:

$$\min_q \ \hat{J}(q) := J(q, u(q)) \tag{3.19}$$

where the weak parabolic side condition (the state equation)

$$e(q, u(q); \varphi) = 0 \tag{3.20}$$

has already been solved and $u(q)$ is the state belonging to the given control q, which can be expressed by means of the solution operator $S : Q \to X,\ q \mapsto S(q) \equiv u(q)$. The first order directional derivative of $\hat{J}$ at the point $q \in Q$ in direction $\delta q \in Q$ is then given by

$$\begin{aligned} \hat{J}'_q(q)(\delta q) &= J'_q(q, u(q))(\delta q) + J'_u(q, u(q))(u'_q(\delta q)) \\ &= \langle J'_q, \delta q \rangle_{Q^* \times Q} + \langle J'_u, u'_q(\delta q) \rangle_{X^* \times X} \\ &= \langle J'_q, \delta q \rangle_{Q^* \times Q} + \langle (u'_q)^*(J'_u), \delta q \rangle_{Q^* \times Q}. \end{aligned} \tag{3.21}$$

The operator $u'_q : Q \to X$ is linear and continuous; its application $u'_q(\delta q)$ to a direction $\delta q \in Q$ is abbreviated by δu and called the sensitivity of u w. r. t. q. In the last abstract representations, the arguments $(q, u(q))$ were omitted, and the adjoint operator $(u'_q)^* : X^* \to Q^*$ was used. The sensitivity δu is obtained abstractly as the solution of a linearized state equation

$$e'_u(\delta u) + e'_q(\delta q) = 0. \tag{3.22}$$

Assuming that the operator e'_u has a bounded inverse, the linear operator u'_q may be expressed as

$$u'_q(\delta q) = -(e'_u)^{-1}[e'_q(\delta q)]. \tag{3.23}$$

We omit the argument δq to obtain the following expressions:

$$u'_q = -(e'_u)^{-1} \circ e'_q \iff u'^*_q = -e'^*_q \circ (e'_u)^{-*}. \tag{3.24}$$

Inserting the last expression into (3.21)results in

$$\hat{J}'_q(q)(\delta q) = \langle J'_q, \delta q\rangle_{Q^*\times Q} - \langle [e'^*_q \circ (e'_u)^{-*}](J'_u), \delta q\rangle_{Q^*\times Q}. \tag{3.25}$$

Now an adjoint variable $z := (e'_u)^{-*}(-J'_u)$ can be defined which is equivalent to z solving the equation

$$e'^*_u(z) + J'_u = 0. \tag{3.26}$$

The latter equation is the so-called adjoint equation in its abstract version. The process for computing and evaluating $\hat{J}'_q(q)(\delta q)$ can now be resumed as follows:

1. Solve the state equation $e(q,u) = 0$ (equation (3.20)) for $u = u(q) \in X$.
2. Solve the adjoint equation $e'^*_u(z) = -J'_u$ (equation (3.26)) for $z = z(u(q)) \in X$. $(\star)$
3. Evaluate $\langle J'_q, \delta q\rangle_{Q^*\times Q} + \langle e'^*_q(z), \delta q\rangle_{Q^*\times Q}$ which is an expression for $\hat{J}'_q(q)(\delta q)$.

As in concrete situations the quantities u and z have to be actually computed, we briefly discuss the concrete equations. The state equation (3.20) corresponds to the side condition (3.6):

$$(\!(\partial_t u, \varphi)\!) + a(u)(\varphi) + b(q)(\varphi) + (u(0), \varphi(0)) = (\!(f, \varphi)\!) + (u_0, \varphi(0)).$$

The reformulation (3.14) shows how to compute $\delta u = u'_q(\delta q)$, namely by solving the linearized equation

$$(\!(\partial_t \delta u, \varphi)\!) + a'_u(u)(\delta u, \varphi) + (\delta u(0), \varphi(0)) = -b'_q(q)(\delta q, \varphi), \tag{3.27}$$

which is obtained by differentiating (3.14) w. r. t. q in direction δq. This corresponds to (3.22). The general adjoint equation of (3.27) is given by

$$-(\!(\partial_t \delta u^*, \psi)\!) + a'_u(u)(\delta u^*, \psi) + (\delta u^*(T), \psi(T)) = \mathrm{rhs}(\psi). \tag{3.28}$$

The righthand side term is given by the u-derivative of the objective functional which leads to the concrete adjoint equation

$$-(\!(\partial_t z, \psi)\!) + a'_u(u)(z, \psi) + (z(T), \psi(T)) = -J'_u(q,u)(\psi). \tag{3.29}$$

This is (3.26) in our example. After solving (3.29), the resulting expression for $\hat{J}'_q(q)(\chi)$ which is given by

$$\hat{J}'_q(q)(\chi) = \alpha(\!(q, \chi)\!) + b'_q(q)(z, \chi) \tag{3.30}$$

can be evaluated. In Chapter 5, the same abstract technique will be used to show the equivalence between two seemingly different approaches to direct multiple shooting. In fact, the two DMS variants will turn out as a non-reduced and a reduced variant, and the reduction will be established by a technically more difficult version of the above abstract argument. We next present an alternative derivation of the adjoint equation which enables us to represent the second derivative of $\hat{J}$ by solving additional linear problems.

The solution of the OCP is known to be among the stationary points of the Lagrange functional

$$\begin{aligned}\mathcal{L}(q,u,z) &= J(q,u) + e(q,u;z) \\ &= J(q,u) + (\!(\partial_t u, z)\!) + a(u)(z) + b(q)(z) - (\!(f,z)\!) + (u(0) - u_0, z(0))\end{aligned} \tag{3.31}$$

which is the sum of the objective functional and the weakly formulated PDE side condition (3.6). For theoretical background, consult the textbooks by Ito & Kunisch [60] or Luenberger [78]. The Lagrange multiplier z denotes the solution of the equation $\mathcal{L}'_u(\xi)(\psi) = 0$ which arises as part of the following optimality conditions (for brevity, $\xi := (q,u,z)$):

$$\mathcal{L}'_z(\xi)(\varphi) = (\!(\partial_t u, \varphi)\!) + a(u)(\varphi) + b(q)(\varphi) - (\!(f,\varphi)\!) + (u(0) - u_0, \varphi(0)) = 0, \tag{3.32a}$$

$$\mathcal{L}'_u(\xi)(\psi) = J'_u(q,u)(\psi) - (\!(\partial_t z, \psi)\!) + a'_u(u)(\psi, z) + (z(T), \psi(T)) = 0, \tag{3.32b}$$

$$\mathcal{L}'_q(\xi)(\chi) = J'_q(q,u)(\chi) + b'_q(q)(\chi, z) = 0. \tag{3.32c}$$

This KKT system consists of the derivatives of (3.31) and reveals the correspondency of (3.6) and (3.32a), of (3.29) and (3.32b), and finally of (3.30) and (3.32c).

Remark 3.8. As predicted in Section 2.4, this system of optimality conditions has the structure of a temporal parabolic BVP. Equation (3.32a) is the state equation to be solved forward in time, and (3.32b) is the adjoint equation to be solved backward in time. The initial values for both components are prescribed at $t = 0$ and $t = T$, respectively. Both equations are additionally coupled by the control equation (3.32c).

It is now explained how the Lagrange formalism can be used to evaluate $\hat{J}'_q(q)(\chi)$ via the adjoint method. If $u = u(q)$, i. e., the side condition has been solved for given q, then we receive the equalities

$$\hat{J}(q) = J(q,u) = \mathcal{L}(\xi). \tag{3.33}$$

For the derivative $\hat{J}'_q(q)(\delta q)$ this means

$$\hat{J}'_q(q)(\delta q) = \mathcal{L}'_q(\xi)(\delta q) + \mathcal{L}'_u(\xi)(\delta u) + \mathcal{L}'_z(\xi)(\delta z), \tag{3.34}$$

where, by the chain rule, $\delta u := u'_q(\delta q)$ and $\delta z := z'_q(\delta q)$. By assumption, the state equation has already been solved, thus $\mathcal{L}'_z(\xi)(\delta z) = 0$ for all $\delta z \in X$. If now the adjoint equation is solved (i. e., $\mathcal{L}'_u(\xi)(\delta u) = 0$ for all $\delta u \in X$), then (3.34) can be written as

$$\hat{J}'_q(q)(\delta q) = \mathcal{L}'_q(q, u(q), z(q))(\delta q) = \alpha(\!(q, \delta q)\!) + b'_q(q)(\delta q, z(q)).$$

Note that the same steps as in $(\star)$ have been carried out. It is straightforward to develop a method for computing the second order directional derivative $\hat{J}''_{qq}(q)(\delta q_2, \delta q_1)$, using the relation (3.33). Equation (3.34) is differentiated to obtain

$$\begin{aligned}\hat{J}''_{qq}(q)(\delta q_2, \delta q_1) = {} & \mathcal{L}''_{qq}(\xi)(\delta q_2, \delta q_1) + \mathcal{L}''_{qu}(\xi)(\delta q_2, \delta u_1) + \mathcal{L}''_{qz}(\xi)(\delta q_2, \delta z_1) \\ & + \mathcal{L}''_{uq}(\xi)(\delta u_2, \delta q_1) + \mathcal{L}''_{uu}(\xi)(\delta u_2, \delta u_1) + \mathcal{L}''_{uz}(\xi)(\delta u_2, \delta z_1) \\ & + \mathcal{L}''_{zq}(\xi)(\delta z_2, \delta q_1) + \mathcal{L}''_{zu}(\xi)(\delta z_2, \delta u_1) + \mathcal{L}''_{zz}(\xi)(\delta z_2, \delta z_1) \\ & + \mathcal{L}'_u(\xi)(\delta u_{12}) + \mathcal{L}'_z(\xi)(\delta z_{12}).\end{aligned} \tag{3.35}$$

Again, the abbreviations $\delta u_i = u'_q(\delta q_i)$ and $\delta z_i = z'_q(\delta q_i)$ are used, and in the nonlinear case also $\delta u_{ji} = u''_{qq}(\delta q_j, \delta q_i)$ and $\delta z_{ji} = z''_{qq}(\delta q_j, \delta q_i)$ are required.

Remark 3.9. Whenever higher order derivatives are involved, the differentiation order has to be read from right to left. For instance, $\mathcal{L}''_{qu}(\xi)(\delta q_2, \delta u_1)$ means that $\mathcal{L}(\xi)$ was first differentiated w. r. t. u in direction δu_1, and then $\mathcal{L}'_u(\xi)(\delta u_1)$ was differentiated w. r. t. q in direction δq_2.

Concerning the representation (3.35) of the second order derivative, the differentiation order commutes as sufficient regularity was assumed for $J(q,u)$ and $e(q,u)$. Furthermore, $\mathcal{L}''_{zz}(\xi)(\cdot,\cdot) \equiv 0$ as $\mathcal{L}'_z(\xi)(\cdot)$ is the state equation which does not depend on z. In the linear case, the last two terms on the righthand side of (3.35) vanish, and in the nonlinear case, they constitute the adjoint and state equations that are assumed as already solved. Thus, only the first eight terms remain to be discussed. Again, there are two approaches to evaluating $\hat{J}''_{qq}(q)(\delta q_2, \delta q_1)$ that can be classified as a sensitivity and an adjoint method. As before, only the adjoint approach is presented, which reflects the implementation of the later examples; Meidner [84] provides a comparative presentation of both approaches.
For given δq, the following equation is solved to obtain δu:

$$\mathcal{L}''_{qz}(\xi)(\delta q, \varphi) + \mathcal{L}''_{uz}(\xi)(\delta u, \varphi) = 0 \quad \forall\, \varphi \in X. \tag{3.36}$$

Once δu is computed, it is used to solve the equation

$$\mathcal{L}''_{qu}(\xi)(\delta q, \psi) + \mathcal{L}''_{uu}(\xi)(\delta u, \psi) + \mathcal{L}''_{zu}(\xi)(\delta z, \psi) = 0 \quad \forall\, \psi \in X. \tag{3.37}$$

for δz. Then the remaining terms of (3.35) constitute an evaluable expression for $\hat{J}''_{qq}(q)(\delta q, \chi)$:

$$\hat{J}''_{qq}(q)(\delta q, \chi) = \mathcal{L}''_{qq}(\xi)(\delta q, \chi) + \mathcal{L}''_{uq}(\xi)(\delta u, \chi) + \mathcal{L}''_{zq}(\xi)(\delta z, \chi). \tag{3.38}$$

Equations (3.36) and (3.37) are called the tangent and the extra adjoint equations, respectively. They both have to be solved in order to evaluate $\hat{J}''_{qq}(q)$ via (3.38). As above for the first order derivative, we state the concrete versions of the tangent and extra adjoint equations in the framework of problem (3.3) subject to (3.6). The tangent equation is given as

$$(\!(\partial_t \delta u, \varphi)\!) + a'_u(u)(\delta u, \varphi) + b'_q(q)(\delta q, \varphi) + (\delta u(0), \varphi(0)) = 0, \tag{3.39}$$

the extra adjoint equation reads

$$-(\!(\partial_t \delta z, \psi)\!) + a'_u(u)(\psi, \delta z) + a''_{uu}(u)(\delta u, \psi, z) + J''_{uu}(q,u)(\delta u, \psi) + (\delta z(T), \psi(T)) = 0, \tag{3.40}$$

and the Hessian of the reduced functional is then evaluated as

$$b'_q(q)(\chi, \delta z) + b''_{qq}(q)(\delta q, \chi, z) + J''_{qq}(q,u)(\delta q, \chi). \tag{3.41}$$

For a linear operator $\mathcal{B}$ and belonging form $b(\cdot)(\cdot)$, the term $b''_{qq}(\cdot)(\cdot,\cdot,\cdot)$ vanishes.

3.4 PDE based OCP and multiple shooting

So far, the solvability as well as conditions for minimal solutions of the OCP (3.3) subject to (3.6) have been discussed. In subsequent chapters, OCP of this type are solved by multiple shooting methods similar to the ones presented in Chapter 2, and therefore the OCP has to be modified. The following reformulation, especially the proof of equivalence, is included in Carraro & Geiger [21]. It relies on a decomposition of the closure $\bar{I}$ of the interval $I = (0, T)$ (cf. (2.15) in the ODE case)

$$\bar{I} = \{\tau_0\} \cup \bigcup_{j=0}^{M-1} I_j, \quad I_j = (\tau_j, \tau_{j+1}]. \tag{3.42}$$

Here, $\tau_0 = 0$ and $\tau_M = T$, and the following redefinition (3.43) of the OCP is done in terms of local control and state functions q^j, u^j on the subintervals I_j. They lie in the intervalwise defined spaces $Q^j := L^2(I_j; R)$ and $X^j := \{v \in L^2(I_j; V) \,|\, \partial_t v \in L^2(I_j; V^*)\}$, respectively. Sometimes, a more global view on these intervalwise problems is required, and therefore the compositions $\mathbf{u} = ((u^j)_{j=0}^{M-1})$ and $\mathbf{q} = ((q^j)_{j=0}^{M-1})$ of the intervalwise states and controls are defined as well as the corresponding spaces

$$\mathbf{X} := \bigtimes_{j=0}^{M-1} X^j, \quad \mathbf{Q} := \bigtimes_{j=0}^{M-1} Q^j.$$

The cross product means that $\mathbf{X} = \{v \in L^2(I; V) \,|\, v|_{I_j} \in X^j\}$ and $\mathbf{Q} = \{q \in L^2(I; R) \,|\, q|_{I_j} \in Q^j\}$ are function spaces on $(0, T)$ consisting of compositions of intervalwise defined functions; for instance, $\mathbf{u} \in \mathbf{X}$ and $\mathbf{q} \in \mathbf{Q}$. This implies $X \subsetneq \mathbf{X}$ and $Q = \mathbf{Q}$, where X and Q are the spaces defined in Section 3.1 for problem (3.3) subject to (3.6). With these notations, the modified (non-reduced) control problem reads:

$$\min_{(\mathbf{q},\mathbf{u})} \bar{J}(\mathbf{q}, \mathbf{u}) := \sum_{j=0}^{M-1} J^j(q^j, u^j) = \frac{\kappa_1}{2} \sum_{j=0}^{M-1} \int_{I_j} \|u^j - \hat{u}|_{I_j}\|_V^2 \, dt$$
$$+ \frac{\kappa_2}{2} \|u^{M-1}(\tau_M) - \hat{u}_T\|_H^2 + \frac{\alpha}{2} \sum_{j=0}^{M-1} \int_{I_j} \|q^j\|_Q^2 \, dt \tag{3.43a}$$

$$\text{s. t.} \quad ((\partial_t u^j, \varphi)) + a(u^j)(\varphi) + b(q^j)(\varphi) - ((f|_{I_j}, \varphi))$$
$$+ (u^j(\tau_j) - s^j, \varphi(\tau_j)) = 0 \quad \text{for} \quad j \in \{0, \ldots, M-1\}. \tag{3.43b}$$

In this formulation, it becomes obvious why $J_1(u)$ has to be decomposable (cf. Remark 3.3). The equations (3.43b) constitute IVP on the subintervals I_j the exact initial values $u(\tau_j)$ of which are unknown. Therefore, artificial initial values $\mathbf{s} = (s^j)_{j=0}^M \in H^{M+1}$ have to be imposed, which leads to jumps in the global solution $\mathbf{u}$ composed of the interval solutions u^j (i. e. $\mathbf{u}|_{I_j} \equiv u^j$). Thus, problem (3.43) cannot be equivalent to the original OCP, because $\mathbf{u} \notin C(\bar{I}; H)$, whereas the solution $u \in X$ of the global OCP has to be continuous on I due to the embedding $X \hookrightarrow C(\bar{I}; H)$ mentioned in Section 3.1. From this it can be seen that the original solution space X is in fact a proper subset of $\mathbf{X}$, as

postulated above. In order to establish equivalence of our modified OCP to the original one, the global continuity of the solution $\mathbf{u}$ of (3.43) has to be enforced by imposing the following additional continuity conditions:

$$(s^0 - u_0, \phi) = 0 \quad \forall\, \phi \in H, \tag{3.44a}$$
$$(s^{j+1} - u^j(\tau_{j+1}), \phi) = 0 \quad \forall\, \phi \in H,\ j \in \{0, \dots, M-1\}. \tag{3.44b}$$

In order to prove the equivalence of the original OCP (3.3) subject to (3.6) and the extended problem (3.43)–(3.44), the following preparatory lemma is required.

Lemma 3.8. *The objective functionals $J(q,u)$ and $\bar{J}(\mathbf{q},\mathbf{u})$ coincide for $\mathbf{u} = ((u^j)_{j=0}^{M-1})$ with $u^j = u|_{I_j}$, i. e. for globally continuous intervalwise defined functions $\mathbf{u}$.*

Proof. Making use of the additivity of integration on subintervals results in

$$J(q,u) = \frac{\kappa_1}{2}\int_I \|u(t) - \hat{u}(t)\|_V^2\, dt + \frac{\kappa_2}{2}\|u(T) - \hat{u}_T\|_H^2 + \frac{\alpha}{2}\int_I \|q(t)\|_R^2\, dt$$
$$= \frac{\kappa_1}{2}\sum_{j=0}^{M-1}\int_{I_j} \|u^j(t) - \hat{u}|_{I_j}(t)\|_V^2\, dt + \frac{\kappa_2}{2}\|u^{M-1}(\tau_M) - \hat{u}_T\|_H^2 + \frac{\alpha}{2}\sum_{j=0}^{M-1}\int_{I_j} \|q^j(t)\|_R^2\, dt.$$

The latter corresponds to $\bar{J}(\mathbf{q},\mathbf{u}) = \sum_{j=0}^{M-1} J^j(q^j, u^j)$. □

Everything is prepared to state the equivalence of the original and the modified OCP unfolding in the following theorem:

Theorem 3.9. *(a) Let $(q,u) \in Q \times X$ be a solution to the original OCP (3.3) subject to (3.6). Then $(\mathbf{q},\mathbf{u}) \in \mathbf{Q} \times \mathbf{X}$, defined by $q^j := q|_{I_j}$ and $u^j := u|_{I_j}$, is a solution to the modified OCP (3.43)–(3.44).*
(b) Let $(\mathbf{q},\mathbf{u}) \in \mathbf{Q} \times \mathbf{X}$ solve the modified problem (3.43)–(3.44). If we define q by $q|_{I_j} := q^j$ and u by $u|_{I_j} := u^j$, then $(q,u) \in Q \times X$ solves the original OCP (3.3) subject to (3.6).

Proof. (a) Since $u \in X$ is globally continuous in time, we have $s^0 = u_0$ as well as $s^{j+1} = u^{j+1}(\tau_{j+1}) = u(\tau_{j+1})$ and $u^j(\tau_{j+1}) = u(\tau_{j+1})$, which means in turn $s^{j+1} = u^j(\tau_{j+1})$. Thus, the matching conditions (3.44) are fulfilled. Let now $(\tilde{\mathbf{q}},\tilde{\mathbf{u}}) = ((\tilde{q}^j, \tilde{u}^j)_{j=0}^{M-1}) \in \mathbf{Q} \times \mathbf{X}$ be given so that $\bar{J}(\tilde{\mathbf{q}},\tilde{\mathbf{u}}) < \bar{J}(\mathbf{q},\mathbf{u})$ and the continuity conditions (3.44) are fulfilled. The latter assumption immediately implies $\tilde{\mathbf{u}} \in X$, i. e. $(\tilde{q},\tilde{u}) := (\tilde{\mathbf{q}},\tilde{\mathbf{u}}) \in Q \times X$ due to $\mathbf{Q} = Q$. Lemma 3.8 now yields

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

which is a contradiction to the assumed optimality of (q,u).
(b) Since $\mathbf{u}$ is part of a solution of the modified OCP, especially (3.44), the initial values are $s^0 = u_0$ and $s^{j+1} = u^j(\tau_{j+1})$. The initial value s^{j+1} on I_{j+1} clearly fulfils $s^{j+1} = u^{j+1}(\tau_{j+1})$. From $\mathbf{u} \in \mathbf{X}$ it is known that $u^j \in C(\bar{I}_j; H)$, and together with the global continuity $\mathbf{u} \in C(\bar{I}; H)$ is guaranteed. Considering $\partial_t u^j \in L^2(I_j; V^*)$, the corresponding global

property $\partial_t u \in L^2(I;V^*)$ directly follows. This means that u, defined by $u|_{I_j} := u^j$, lies in X, and together with q (analogously defined by $q|_{I_j} := q^j$) one obtains $(q,u) \in Q \times X$. Assuming that there is $(\tilde{q},\tilde{u}) \in Q \times X$ with $J(\tilde{q},\tilde{u}) < J(q,u)$, the contradiction

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

to the optimality of $(\mathbf{q},\mathbf{u})$ is obtained by Lemma 3.8. □

The reformulated problem (3.43)–(3.44) is a suitable starting point for the multiple shooting algorithms that are presented in detail in Chapter 5. As stated in Section 3.3, solution algorithms for OCP are often based on first and second order optimality conditions. Thus, to present the indirect and direct multiple shooting methods properly, we have to derive at least the first order necessary optimality conditions of the modified OCP. Therefore, the corresponding Lagrange functional is defined, which is an extended version of (3.31) where the additional equality constraints (3.44) are taken into account. This extended Lagrangian is given as follows:

$$\begin{aligned}\bar{\mathcal{L}}\left((q^j,u^j,z^j)_{j=0}^{M-1},(s^j,\lambda^j)_{j=0}^{M}\right) := & \sum_{j=0}^{M-1} J^j(q^j,u^j) \\ & + \sum_{j=0}^{M-1}\Big[(\!(\partial_t u^j,z^j)\!) \;+\; a(u^j)(z^j) + b(q^j)(z^j) - (\!(f|_{I_j},z^j)\!)\Big] \\ + \sum_{j=0}^{M-1}(u^j(\tau_j)-s^j,z^j(\tau_j)) \;+\; & \sum_{j=0}^{M-1}(s^{j+1}-u^j(\tau_{j+1}),\lambda^{j+1}) + (s^0-u_0,\lambda^0).\end{aligned} \tag{3.47}$$

The first line of (3.47) contains the modified cost functional (3.43a) which has been rearranged in an intervalwise fashion. In the second line, the intervalwise parabolic side conditions (3.43b) are included without the initial conditions. They, together with the continuity conditions (3.44), form the third line of (3.47), where all terms containing the variables s^j are gathered. There are two kinds of Lagrange multipliers: the adjoint variables $\mathbf{z} = ((z^j)_{j=0}^{M-1}) \in \mathbf{X}$ corresponding to the intervalwise PDE side condition, and the spatial functions $\boldsymbol{\lambda} = (\lambda^j)_{j=0}^{M} \in H^{M+1}$ as multipliers for the equality constraints (3.44). Now the first order optimality conditions, or KKT system, can be derived by differentiating the above Lagrangian w. r. t. all its arguments. This yields, with the abbreviation $\xi = ((q^j,u^j,z^j)_{j=0}^{M-1},(s^j,\lambda^j)_{j=0}^{M})$, for all test functions $(\delta z,\delta u,\delta q,\delta\lambda,\delta s) \in X^j \times X^j \times Q^j \times H \times H$

and for all $j \in \{0, \cdots, M-1\}$, the intervalwise equations

$$\begin{aligned}
\bar{\mathcal{L}}'_{z^j}(\xi)(\delta z) = (\!(\partial_t u^j, \delta z)\!) + a(u^j)(\delta z) + b(q^j)(\delta z) & \\
-(\!(f|_{I_j}, \delta z)\!) + (u^j(\tau_j) - s^j, \delta z(\tau_j)) &= 0, \quad (3.48a)\\
\bar{\mathcal{L}}'_{u^j}(\xi)(\delta u) = J_u^{j\prime}(q^j, u^j)(\delta u) - (\!(\partial_t z^j, \delta u)\!) + a'_u(u^j)(\delta u, z^j) & \\
+(z^j(\tau_{j+1}) - \lambda^{j+1}, \delta u(\tau_{j+1})) &= 0, \quad (3.48b)\\
\bar{\mathcal{L}}'_{q^j}(\xi)(\delta q) = J_q^{j\prime}(q^j, u^j)(\delta q) + b'_q(q^j)(\delta q, z^j) &= 0, \quad (3.48c)\\
\bar{\mathcal{L}}'_{\lambda^0}(\xi)(\delta\lambda) = (s^0 - u_0, \delta\lambda) &= 0, \quad (3.48d)\\
\bar{\mathcal{L}}'_{\lambda^j}(\xi)(\delta\lambda) = (s^{j+1} - u^j(\tau_{j+1}), \delta\lambda) &= 0, \quad (3.48e)\\
\bar{\mathcal{L}}'_{s^j}(\xi)(\delta s) = (\lambda^j - z^j(\tau_j), \delta s) &= 0, \quad (3.48f)\\
\bar{\mathcal{L}}'_{s^M}(\xi)(\delta s) = (\lambda^M, \delta s) &= 0. \quad (3.48g)
\end{aligned}$$

Remark 3.10. For the KKT system, the cases $\kappa_1 = 1,\ \kappa_2 = 0$ and $\kappa_1 = 0,\ \kappa_2 = 1$ have to be distinguished. In the above notation, this does only affect the adjoint equation (3.48b). In case of a distributed objective functional, all subintervals can be treated equally, and it holds $J_u^{j\prime}(q^j, u^j)(\delta u) = (\!(u^j - \hat{u}|_{I_j}, \delta u)\!)$ for all $j \in \{0, \cdots, M-1\}$. In case of an end-time functional term, one obtains $J_u^{j\prime}(q^j, u^j)(\delta u) \equiv 0$ for $j \in \{0, \cdots, M-2\}$, and $J_u^{M-1\prime}(q^{M-1}, u^{M-1})(\delta u) = (u^{M-1}(\tau_M) - \hat{u}_T, \delta u(\tau_M))$.

Agreement. *If not specified differently, further on $\kappa_1 = 1$ and $\kappa_2 = 0$ is assumed for the theoretical considerations, corresponding to a distributed objective functional. In this case, the presentation is uniform on all subintervals and the necessary modifications in case of an end-time functional are straightforward.*

This system of equations can be split into two parts. The first one, equations (3.48a)–(3.48c), corresponds to the KKT system of the original problem (3.3) subject to (3.6), but restricted to a subinterval I_j (compare these equations to (3.32)). The corresponding unknowns u^j, z^j and q^j are functions depending on spatial variables and time. The second part consists of equations (3.48d)–(3.48g) and appears in the modified problem (3.43)–(3.44). The unknowns s^j and λ^j are spatial functions at the isolated time-points τ_j and do not depend on time t.
Stationary points of the Lagrangian, i. e., solutions of (3.48), are solution candidates for the modified OCP. The KKT system constitutes a root-finding problem which can be handled by Newton's method. For this purpose, the second order derivatives of the extended Lagrange functional (3.47) are required which constitute the Jacobian of the optimality conditions (3.48). Although the algorithmic details are explained later (for a presentation of Newton's method, see Section 4.3, whereas details of the different multiple shooting processes are explained in Sections 5.1 and 5.2), we conclude the current chapter by providing the background for these later discussions. Therefore, it is briefly recalled that Newton's method for solving a nonlinear but continuously differentiable problem $f(x) = 0$ consists in the iteration

$$x_{k+1} = x_k - J_f(x_k)^{-1} f(x_k),$$

initialized by a suitable starting point x_0. To avoid the expensive explicit inversion of the Jacobian J_f, this is usually written in the two-step form

$$J_f(x_k)\delta x = -f(x_k), \tag{3.49a}$$
$$x_{k+1} = x_k + \delta x. \tag{3.49b}$$

The linear system displayed in the following is a formal representation of (3.49a) transferred to our context. The equations (3.48) are rearranged in a way that facilitates the illustration of IMS and DMS concepts in Chapter 5 (for instance, equations (3.48d) and (3.48e) as well as (3.48f) and (3.48g) have been resumed symbolically as $\bar{\mathcal{L}}'_\lambda$ and $\bar{\mathcal{L}}'_s$, respectively). The resulting system reads

$$\begin{pmatrix} 0 & \bar{\mathcal{L}}''_{uz} & \bar{\mathcal{L}}''_{qz} & \bar{\mathcal{L}}''_{sz} & 0 \\ \bar{\mathcal{L}}''_{zu} & \bar{\mathcal{L}}''_{uu} & 0 & 0 & \bar{\mathcal{L}}''_{\lambda u} \\ \bar{\mathcal{L}}''_{zq} & 0 & \bar{\mathcal{L}}''_{qq} & 0 & 0 \\ \bar{\mathcal{L}}''_{zs} & 0 & 0 & 0 & \bar{\mathcal{L}}''_{\lambda s} \\ 0 & \bar{\mathcal{L}}''_{u\lambda} & 0 & \bar{\mathcal{L}}''_{s\lambda} & 0 \end{pmatrix} \begin{pmatrix} \delta z \\ \delta u \\ \delta q \\ \delta s \\ \delta\lambda \end{pmatrix} = - \begin{pmatrix} \bar{\mathcal{L}}'_z \\ \bar{\mathcal{L}}'_u \\ \bar{\mathcal{L}}'_q \\ \bar{\mathcal{L}}'_s \\ \bar{\mathcal{L}}'_\lambda \end{pmatrix}. \tag{3.50}$$

The righthand side of (3.50) consists of block vectors. The components of $\bar{\mathcal{L}}'_z$, e. g., are the subinterval state equations, i. e., $\bar{\mathcal{L}}'_z = (\bar{\mathcal{L}}'_{z^0}, \cdots, \bar{\mathcal{L}}'_{z^{M-1}})^\top$. Analogously, each of the sensitivity (or solution) variables is a block vector consisting of subinterval sensitivities (e. g., $\delta q = (\delta q^{(0)}, \cdots, \delta q^{(M-1)})^\top$). The blocks of the matrix are either zero submatrices in case the equation to be differentiated does not depend on the variable with respect to which we differentiate, or they are sparse (often diagonal) matrices due to the decoupling of the component equations of (3.48) between different subintervals. In the context of parabolic OCP this system is never assembled explicitly due to its size. The different multiple shooting techniques rely on different splittings of system (3.50) which reduce its size significantly. Nevertheless, the corresponding smaller matrices are still not assembled. Instead, Krylov-Newton methods are employed that allow to solve the respective Newton equations in a matrix-free manner (see Sections 4.2 and 4.3 for details).

Remark 3.11. For a discussion of the size of system (3.50), we exemplarily describe the matrix in more detail. The upper left 3×3 block consists of nine quadratic $M \times M$ blocks, whereas the lower right 2×2 block comprises four $(M+1) \times (M+1)$ blocks. The remaining submatrices are rectangular matrices of appropriate dimension. Summarizing, the Newton matrix is of size $(5M+2) \times (5M+2)$. Assuming the number M of subintervals I_j to be of moderate size (from $M = 1$ in the case of simple shooting up to multiple shooting with $M \approx 10 - 100$), the system appears to be small. However, system (3.50) still describes a function space environment, i. e., neither time nor space discretization are considered so far. Chapters 4 and 5 demonstrate that especially the discretization of the spatial variables leads to a huge enlargement of the systems that have to be solved numerically.

4 Discretization and Solvers

In the previous chapter, the theoretical framework of parabolic OCP was presented and adapted to the multiple shooting context. The aim of the current chapter is to present numerical tools in order to prepare the numerical tests for the methods developed in Chapters 5–7 which confirm the theoretical findings. By that, details of implementation are discussed which enlighten the numerical results presented in Chapter 2 in the ODE context.
We address three main topics: In Section 4.1, several discretization schemes for the temporal, the spatial, and the control variables are discussed and their most important numerical properties are recapitulated. Section 4.2 is concerned with iterative solution schemes for linear equation systems. The conjugate gradient and generalized minimum residual algorithms are introduced, thereby emphasizing their matrix-free realization. These linear solvers constitute an essential part of the implementation underlying the numerical examples in later chapters. Finally, an overview on some variants of Newton's method is presented in Section 4.3, as this method appears in several forms throughout this thesis. Inexact Newton approaches, especially Krylov-Newton algorithms, are covered that are intertwined with the iterative linear solvers from Section 4.2.

4.1 Discretization

There are three types of variables to be discretized in the framework of nonstationary PDE governed optimal control problems. First, the discretization of the time interval $I = (0, T]$, respectively the multiple shooting subintervals $I_j = (\tau_j, \tau_{j+1}]$ (see (2.15)), is performed either by one-step difference schemes or by a discontinuous or continuous Galerkin method. Both concepts and their connection are presented in Subsection 4.1.1. Second, the discretization of the spatial domain $\Omega \subset \mathbb{R}^d$ is carried out by a conforming finite element method; the details are given in Subsection 4.1.2. Lastly, possible discretization approaches for the control variable are presented in Subsection 4.1.3. The topic of control parameterization is revisited and complemented with a justification why, in the PDE case, a full discretization is preferred.

4.1.1 Discretization of the time variable

It is important to point out the difference between time discretization methods presented in this section and time domain decomposition methods based on the splitting (3.42) of the solution interval I. In Section 3.4, the latter led to an extended problem formulation that

is fully equivalent to the original one. In contrast, the following discretization methods modify the given continuous problem. They aim at producing a series of finite dimensional problems that approximate the original problem, so that the discrete solutions converge to the continuous solution as a given discretization parameter converges to zero. Time domain decomposition methods, such as multiple shooting, can be formulated on the continuous level within an abstract function space setting.

The time semi-discretization relies on partitioning the closed shooting intervals $\bar{I}_j = \{\tau_j\} \cup (\tau_j, \tau_{j+1}]$ given in (3.42) into smaller subintervals $I_n^j = (t_{n-1}^j, t_n^j]$ of length k_n^j with left and right end points

$$\tau_j = t_0^j < t_1^j < \cdots < t_{N_j}^j = \tau_{j+1}. \tag{4.1}$$

The temporally semi-discrete variables are indicated by a subscript k (e. g., the semi-discrete state and adjoint on I_j read u_k^j and z_k^j, respectively). This refers to a piecewise constant step-size function $k(t)$ defined by $k|_{I_n^j} := k_n^j$. For simplicity of notation, the forms $(\!(\cdot,\cdot)\!)_n, a_n(\cdot)(\cdot)$ and $b_n(\cdot)(\cdot)$ are always defined on the subintervals I_n^j as can be seen by their respective arguments.

Difference methods. One-step difference methods aim at computing the approximative solution value $u_{k,n}^j \approx u^j(t_n^j)$ from the preceding value $u_{k,n-1}^j$; the initial value $u_{k,0}^j$ has to be prescribed. They can be derived by approximating temporal derivatives by means of difference quotients. In the practical implementation, standard instantiations of the θ-method

$$u_{k,n}^j = u_{k,n-1}^j + k_n^j \left(\theta F(t_n^j) + (1-\theta) F(t_{n-1}^j)\right) \tag{4.2}$$

are employed. The term $F(\bar{t})$ comprises the evaluation of all problem parts except the time derivative at the fixed timepoint $\bar{t}$. The choice $\theta = 1$ yields the backward Euler method which is implicit and therefore involves the solution of a nonlinear equation per time step. The backward Euler scheme is of first order, i. e., $|u(t) - u_k(t)|_2 = \mathcal{O}(k)$. Taking $\theta = 0.5$ leads to the semi-implicit Crank-Nicolson scheme, which is of second order, i. e., $|u(t) - u_k(t)|_2 = \mathcal{O}(k^2)$. The Crank-Nicolson scheme is used for discretizing the ODE examples in Chapters 2 and 7, whereas in the PDE context later on, the backward Euler scheme is applied exclusively. Although this method is only of first order, it is straightforward to implement and has further advantages included in the discussion below.

Remark 4.1. In the ODE context, and for PDE governed control problems in the method of lines (MOL) framework, linear multistep methods are a widely used alternative (see, e. g., Beigel [8] and the references therein). The implicit backward difference formulae (BDF) are suitable for solving stiff problems. The simplest BDF method coincides with the backward Euler scheme.

Galerkin methods. In contrast to the difference schemes, Galerkin methods are based on the variational problem formulation (see (3.6) or the intervalwise formulation (3.43b)).

To introduce the discontinuous Galerkin method of order $r \geq 0$ (in short: 'dG(r) method'), the semi-discrete space

$$X_k^r(I_j) := \{v_k^j \in L^2(I_j, H) \,|\, v_k^j(t_0^j) \in H, v_k^j|_{I_n^j} \in P_r(I_n^j, V), n = 1, \ldots, N_j\}, \tag{4.3}$$

is introduced, where P_r is the space of polynomials up to degree r defined on the intervals I_n^j. Accordingly, $X_k^r(I_j)$ denotes the space of piecewise polynomial functions on I_j, in general discontinuous between two intervals I_n^j of the time discretization. The following standard notation is required to formulate the weak formulation of the problems:

$$v_{k,n}^{j,+} = \lim_{t \to 0^+} v_k(t_n^j + t), \quad v_{k,n}^{j,-} = \lim_{t \to 0^-} v_k(t_n^j + t), \quad [v_k^j]_n = v_{k,n}^{j,+} - v_{k,n}^{j,-}.$$

Then, the semi-discretization in time of the state equation (3.48a) seeks $u_k^j \in X_k^r(I_j)$ and $s^j \in H$, for $j = 0, \ldots, M-1$, such that

$$\begin{aligned} \sum_{n=1}^{N_j} \Big[(\!(\partial_t u_k^j, \delta z_k)\!)_n + a_n(u_k^j)(\delta z_k) + b_n(q^j)(\delta z_k) - (\!(f|_{I_n^j}, \delta z_k)\!)_n \Big] \\ + \sum_{n=2}^{N_j} ([u_k^j]_{n-1}, \delta z_{k,n-1}^+)) + (u_k^j(\tau_j) - s^j, \delta z_k(\tau_j)) = 0 \end{aligned} \tag{4.4}$$

holds for all $\delta z_k \in X_k^r(I_j)$. Analogously, the semi-discretization of the adjoint equation (3.48b) seeks $u_k^j, z_k^j \in X_k^r(I_j)$ and $\lambda^{j+1} \in H$, for $j = 0, \ldots, M-1$, such that

$$\begin{aligned} J_{u^j}^{j\prime}(q^j, u_k^j)(\delta u_k) - \sum_{n=1}^{N_j} \Big[(\!(\partial_t z_k^j, \delta u_k)\!)_n - a'_{n,u^j}(u_k^j)(\delta u_k, z_k^j) \Big] \\ - \sum_{n=1}^{N_j - 1} ([z_k^j]_n, \delta u_{k,n}^-)) + (z_k^j(\tau_{j+1}) - \lambda^{j+1}, \delta u_k(\tau_{j+1})) = 0 \end{aligned} \tag{4.5}$$

holds for all $\delta u_k \in X_k^r(I_j)$. Thereby, a distributed functional $J^j(q^j, u^j)$ is assumed, and necessary modifications in case of an end-time functional are accounted for.

Equivalently to the discretization of the state and adjoint equations, one needs to discretize the tangent and the additional adjoint equations, respectively. These are not presented here, and we refer to Becker, Meidner & Vexler [7] for further details on this topic in the context of optimal control without multiple shooting. The relevant changes of these discretizations in the multiple shooting context are similar to the ones shown above for the state and adjoint equations.

Remark 4.2. For $r = 0$ the dG(0) method can be interpreted as the classical backward Euler time-stepping method if the occurring time integrals are evaluated by the box rule, i. e., $\int_a^b f(t)\mathrm{d}t \approx (b-a)f(b)$. From this equivalence it can be inferred that the dG(0) method converges with order one.

Although it is not used in our implementation, we comment on the continuous Galerkin method of order $r \geq 1$, as it is equivalent to the Crank-Nicolson method in the case $r = 1$

(see Remark 4.3 below). It uses continuous trial functions of polynomial degree r and discontinuous test functions of polynomial degree $r-1$. The test functions are taken from a space of type (4.3), while for the trial functions, the space

$$Y_k^r(I_j) := \{v_k^j \in C(\overline{I}_j, H) \,|\, v_k^j|_{I_n^j} \in P_r(I_n^j, V), n = 1, \ldots, N_j\},$$

is appropriate, where here P_r is the space of polynomials up to degree $r \geq 1$ defined on the intervals I_n^j. Then, the discrete state equation is formulated as follows: For $j = 0, \ldots, M-1$ find a state $u_k^j \in Y_k^r(I_j)$ such that

$$\sum_{n=1}^{N_j} \left[((\partial_t u_k^j, \delta z_k))_n + a_n(u_k^j)(\delta z_k) + b_n(q^j)(\delta z_k) - ((f|_{I_n^j}, \delta z_k))_n \right] + (u_k^j(\tau_j) - s^j, \delta z_k(\tau_j)) = 0$$

holds for all $\delta z_k \in X_k^{r-1}(I_j)$. Analogously, the adjoint problem reads: For $j = 0, \ldots, M-1$ find an adjoint state $z_k^j \in X_k^{r-1}(I_j)$ such that

$$J_{u^j}^{j\prime}(q^j, u_k^j)(\delta u_k) - \sum_{n=1}^{N_j} \left[((\partial_t z_k^j, \delta u_k))_n - a'_{n,u^j}(u_k^j)(\delta u_k, z_k^j) \right] - \sum_{n=1}^{N_j-1} ([z_k^j]_n, \delta u_{k,n}^-)) + (z_k^j(\tau_{j+1}) - \lambda^{j+1}, \delta u_k(\tau_{j+1})) = 0$$

holds for all $\delta u_k \in Y_k^r(I_j)$. Note that the adjoint equation comprises jump terms, as its solution variable z_k^j comes from a space of globally discontinuous functions.

Remark 4.3. The choice $r = 1$ and the trapezoidal rule as quadrature rule (i. e., $\int_a^b f(t)\mathrm{d}t \approx (b-a)/2\,[f(a) + f(b)]$) lead to the Crank-Nicolson method, which is the time-stepping method corresponding to the cG(1) scheme. Thus, the cG(1) scheme is quadratically convergent.

Remark 4.4. Although the Galerkin methods introduced in this subsection are more complicated compared to the corresponding difference schemes, they are advantageous in two important respects. Their employment guarantees equivalence of 'discretize-then-optimize' and 'optimize-then-discretize' (see Meidner [84]) and they facilitate the formulation of a temporally adaptive dual weighted residual (DWR) method. Thus, even if the corresponding one-step difference methods are preferred in the implementation, theoretical issues on the discrete level are mostly presented in the Galerkin framework. Moreover, this allows to treat temporal and spatial discretization similarly.

4.1.2 Discretization of the spatial variables

The temporally discrete formulation is still continuous w. r. t. the spatial variables x, i. e., elements u_k^j of the space $X_k^r(I_j)$ still map time t into the continuous space V. Henceforth,

dG(r) methods are in the focus. The spatial discretization by means of the finite element method (FEM) replaces V by a discrete function space V_h; we always work with conforming finite elements, i. e., $V_h \subset V$ is further on assumed. More details on the following presentation as well as alternative approaches are presented in the textbooks of Braess [17], Brenner & Scott [18] or Ciarlet [24].
In the examples, the spatial domain Ω is a two-dimensional polygonal set; curved boundaries are handled in the mentioned literature, and the extension to three space dimensions is straightforward. A partition of Ω into open quadrilaterals K, the so-called cells, is considered. The diameters h_K of the cells define a piecewise constant mesh-size function $h(x)$ defined by $h_{|K} := h_K$. Sometimes, an ambiguous notation is used and h is interpreted as the maximum diameter, $h = \max_K h_K$. The resulting triangulation is denoted by $\mathcal{T}_h$; in general, quantities related to the discretization are denoted by a subscript h. The boundary of each cell K consists of straight line segments, the so-called edges.

Definition 4.1. *A triangulation $\mathcal{T}_h = \{K\}$ is regular if the following conditions hold:*

(i) $\bar{\Omega} = \bigcup_{K \in \mathcal{T}_h} \bar{K}$,

(ii) different cells do not overlap, i. e., for $i \neq j$ it holds $K_i \cap K_j = \emptyset$,

(iii) the intersection of two different closed cells $\bar{K}_i$ and $\bar{K}_j$ is either empty, or it consists of a corner point of both $\bar{K}_i$ and $\bar{K}_j$ or a whole edge belonging to both $\bar{K}_i$ and $\bar{K}_j$.

On the regular mesh $\mathcal{T}_h$, the conforming finite element space $V_h^s \subset V$ is defined as a finite dimensional space of piecewise polynomial functions,

$$V_h^s := \{v_h \in V \cap C(\bar{\Omega}) \mid v_{h_{|K}} \in Q^s(K), K \in \mathcal{T}_h\}. \tag{4.6}$$

In order to specify the set $Q^s(K)$ with $s \in \mathbb{N}$ as a space of polynomial-like functions on $K \in \mathcal{T}_h$, the space $\hat{Q}^s(\hat{K})$ of polynomial functions is defined on the reference cell $\hat{K} := (0,1)^2$. It holds

$$\hat{Q}^s(\hat{K}) := \operatorname{span}\left\{x_1^{\alpha_1} x_2^{\alpha_2} \mid \alpha_i \in \{0, 1, \ldots, s\}\right\}. \tag{4.7}$$

From this, the space $Q^s(K) = \{v : K \to \mathbb{R} \mid v \circ \mathcal{T}_K \in \hat{Q}^s(\hat{K})\}$ is obtained by means of the transformations $\mathcal{T}_K : \hat{K} \to K$. If the transformation $\mathcal{T}_K$ is of the same polynomial type as the functions on the reference cell, the resulting finite element is called isoparametric. Following this, the function space for the full space-time discretization is defined as:

$$X_{h,k}^{s,r}(I_j) := \{v_{hk}^j \in L^2(I_j, H) \,|\, v_{hk}^j(t_0^j) \in V_h^s, v_{hk}^j|_{I_n^j} \in P_r(I_n^j, V_h^s), n = 1, \ldots, N_j\}. \tag{4.8}$$

Due to the conformity of $V_h^s \subset V$, the inclusion $X_{h,k}^{s,r}(I_j) \subset X_k^r(I_j)$ holds. The fully discrete functions v_{hk} are provided with a double subscript indicating that both space and time variables are discretized. The resulting method is known in the literature (see, e. g., Eriksson et al. [38]) as the cG(s)dG(r) method, where the first part denotes the continuous spatial discretization and the second part refers to the discontinuous time discretization scheme. In later examples, the cG(1)dG(0) method is employed.

The full space-time discretization for the state equation seeks $u_{hk}^j \in X_{h,k}^{s,r}(I_j)$ and $s_h^j \in V_h^s$, for $j = 0, \ldots, M-1$, such that

$$\begin{aligned}\sum_{n=1}^{N_j} \Big[(\!(\partial_t u_{hk}^j, \delta z_{hk})\!)_n + a_n(u_{hk}^j)(\delta z_{hk}) + b_n(q^j)(\delta z_{hk}) - (\!(f|_{I_n^j}, \delta z_{hk})\!)_n \Big] \\ + \sum_{n=2}^{N_j} ([u_{hk}^j]_{n-1}, \delta z_{hk,n-1}^+)) + (u_{hk}^j(\tau_j) - s_h^j, \delta z_{hk}(\tau_j)) = 0\end{aligned} \tag{4.9}$$

holds for all $\delta z_{hk} \in X_{h,k}^{s,r}(I_j)$. Analogously, the full discretization of the adjoint equation seeks $u_{hk}^j, z_{hk}^j \in X_{h,k}^{s,r}(I_j)$ and $\lambda_h^{j+1} \in V_h^s$, for $j = 0, \ldots, M-1$, such that

$$\begin{aligned}J_{u^j}^{j\prime}(q^j, u_{hk}^j)(\delta u_{hk}) - \sum_{n=1}^{N_j} \Big[(\!(\partial_t z_{hk}^j, \delta u_{hk})\!)_n - a'_{n,u^j}(u_{hk}^j)(\delta u_{hk}, z_{hk}^j) \Big] \\ - \sum_{n=1}^{N_j-1} ([z_{hk}^j]_n, \delta u_{hk,n}^-)) + (z_{hk}^j(\tau_{j+1}) - \lambda_h^{j+1}, \delta u_{hk}(\tau_{j+1})) = 0\end{aligned} \tag{4.10}$$

holds for all $\delta u_{hk} \in X_{h,k}^{s,r}(I_j)$. Apart from the additional subscript h in the fully discrete case, both formulations (4.4)–(4.5) and (4.9)–(4.10) are identical. Thus, the fully discrete formulation (4.9)–(4.10) constitutes a finite dimensional system of equations, whereas the semi-discrete equations (4.4)–(4.5) are still infinite dimensional. In both formulations, the intervalwise control q^j is left unchanged, i. e., the control is not discretized yet; concepts of control discretization are briefly discussed below in Subsection 4.1.4.

4.1.3 Static and dynamic discretization

In the following, the interplay between temporal and spatial discretizationis discussed and the concepts of static, piecewise static and dynamic discretization are introduced. These concepts become important in the framework of adaptive mesh refinement. The case of dynamic spatial meshes is thoroughly examined in Schmich [103] and Schmich & Vexler [104] and employed in Meidner [84], whereas the idea of piecewise static spatial meshes appears in Hesse & Kanschat [53]. In all cases described below, the time grid is fixed before discretizing the spatial variables. This reflects the above proceeding of discretizing first in time and then in space and corresponds to Rothe's method (in contrast to the method of lines (MOL) which discretizes in reverse order).

Static space discretization. A fixed spatial mesh is chosen and used in every time step. Usually this static mesh is a globally refined standard mesh consisting of congruent square cells, but in principle one could also employ a locally refined mesh as long as it does not change over time. This approach is easy to implement, but prohibits any adaptation of the spatial mesh to dynamically changing features of the solution. Nevertheless, the static approach is used in the numerical examples of Chapters 5–7.

Piecewise static space discretization. It is suggestive to think of a piecewise static approach in the context of multiple shooting. Using the structure given by the time domain

decomposition (3.42), a different spatial mesh can be prescribed on each shooting interval. Within the shooting interval, this spatial mesh is kept fixed over time. In contrast to the globally static approach, dynamic changes of the solution can be accounted for at least at the shooting points τ_j.
This idea is also realized outside a multiple shooting framework by freezing one spatial mesh for a given number of time steps and then replacing it by a different one. Note that at timepoints where the spatial mesh changes, the computability of the corresponding solution functions has to be guaranteed. This is usually done by merging both adjacent meshes and considering finite element spaces on the common refinement (see, e. g., Schmich [103]); the latter presumes that all spatial meshes can be derived from some common basic mesh, i. e., hierarchical meshes are considered.

Dynamic space discretization. The fully discrete problem formulation of Subsection 4.1.2, especially the definition of the discrete space $X_{h,k}^{s,r}(I_j)$ in (4.8), is tailored to the static space discretization, as the conforming finite element space V_h^s is chosen for a fixed spatial mesh $\mathcal{T}_h$. To capture dynamic changes of the solution in Ω, one can choose a corresponding spatial mesh $\mathcal{T}_h^n$ and belonging discrete spaces $V_h^{s,n}$ for each timepoint t_n^j. Then the definition of the cG(s)dG(r) trial and test space is modified to

$$X_{h,k}^{s,r}(I_j) := \{v_{hk}^j \in L^2(I_j, H) \,|\, v_{hk}^j(t_0^j) \in V_h^{s,0}, v_{hk}^j|_{I_n^j} \in P_r(I_n^j, V_h^{s,n}), n = 1, \ldots, N_j\}. \tag{4.11}$$

The rest of the problem formulation remains unchanged, especially equations (4.9)–(4.10) can be directly adopted. For dynamic space discretization, the process of merging two successive spatial meshes and working with a common refinement has to be performed at every timepoint, which makes the implementation harder and the computation more costly. Notably, for cG(r) time discretization, the dynamic space discretization becomes more difficult, since then the global temporal continuity across the different spatial meshes has to be ensured; details can be found in Meidner [84].

4.1.4 Control discretization

So far, the control functions are undiscretized (cf. (4.9)–(4.10)). The control space has been introduced in Section 3.1 as $Q = L^2(I; R)$ with a spatial Banach space R. Later, the analogous intervalwise spaces $Q^j = L^2(I_j; R)$ and their product $\mathbf{Q}$ have been defined, and it was proved that $\mathbf{Q} \equiv Q$ (cf. Section 3.4). There are several ways of discretizing q^j; judged by implementational simplicity, the most suitable one is to choose the same time grids and the same spatial meshes for the control and the state and adjoint variables. In this case, no additional interpolation or grid transfer is necessary and the discrete control q_{hk}^j is as finely resolved as the solution variables u_{hk}^j and z_{hk}^j. This is the discretization chosen in later chapters, and the corresponding discrete control space is denoted by Q_d^j.
Hesse [52] suggests on the continuous level a choice of intervalwise constant controls, i. e., $q^j \equiv q_{hk}^j = c_j$ with temporally constant functions $c_j \in R$. This choice of Q amounts to a special case of control parameterization. In Subsection 2.3.3 (cf. also Example 2.2), this was shown to lead to suboptimal solutions; in ODE optimal control, parameterized controls

often yield remarkable results. However, in the PDE framework, parameterization of this kind leads to an information loss on the spatial development of q^j over time. Therefore, the concept of parameterized controls is not pursued any further.
Another reduction of the control space is sketched by Meidner [84]; the idea is briefly described as it marks a trade-off between the full discretization and the rejected parameterization concept.
Let u and z be discretized by a cG(s)dG(r) method on a time grid $\mathcal{I}_k = \{t_n\}_{n=0}^N$ and a spatial mesh $\mathcal{T}_h$. For discretizing q, one can select a different time grid or spatial mesh; they can be chosen independently from $\mathcal{I}_k$ and $\mathcal{T}_h$. As this requires a sophisticated mesh management, one often simply chooses hierarchical coarsenings $\mathcal{I}_{2k}$ or $\mathcal{T}_{2h}$ of the original discretizations. Alternatively, keeping $\mathcal{I}_k$ and $\mathcal{T}_h$, one could work with Galerkin methods of lower order for the control, i. e., if $\tilde{s} < s$ and $\tilde{r} < r$, the control could be discretized by a cG($\tilde{s}$)dG($\tilde{r}$) method. This resembles the usual treatment of variables in fluid mechanics, where the pressure p is often discretized by a lower order finite element than the velocity $\mathbf{v}$, as equal-order finite elements lack the crucial property of inf-sup stability.

Remark 4.5. All suggested concepts discretize the control explicitly, i. e., partitions of both I_j and Ω as well as belonging function spaces for q^j are given in advance. Hinze [58] introduces an implicit control discretization, which relies on the relation between adjoint state and control given by the first order optimality conditions. He shows optimal order convergence in the case of linear-quadratic elliptic OCP with and without additional control constraints. This approach is also presented in detail in the textbook by Hinze et al. [59].

Let us finally state the fully discrete OCP with multiple shooting. With the above suggested discrete control space $Q_d^j \subset Q^j$, the spaces

$$\mathbf{X}_{h,k}^{s,r} := \bigtimes_{j=0}^{M-1} X_{h,k}^{s,r}(I_j), \quad \mathbf{Q}_d := \bigtimes_{j=0}^{M-1} Q_d^j$$

are defined, leading to the following discrete problem (where $\varphi_{hk} \in \mathbf{X}_{h,k}^{s,r}$):

$$\begin{aligned} &\min_{\mathbf{q}_{hk}\in\mathbf{Q}_d, \mathbf{u}_{hk}\in\mathbf{X}_{h,k}^{s,r}} \bar{J}(\mathbf{q}_{hk}, \mathbf{u}_{hk}) \\ \text{s. t.} \quad &\sum_{n=1}^{N_j} \Big[(\!(\partial_t u_{hk}^j, \varphi_{hk})\!)_n + a_n(u_{hk}^j)(\varphi_{hk}) + b_n(q_{hk}^j)(\varphi_{hk}) - (\!(f|_{I_n^j}, \varphi_{hk})\!)_n \Big] \\ &\qquad + \sum_{n=2}^{N_j} ([u_{hk}^j]_{n-1}, \varphi_{hk,n-1}^+)) + (u_{hk}^j(\tau_j) - s_h^j, \varphi_{hk}(\tau_j)) = 0 \quad \forall\, j \in \{0, \dots, M-1\}. \end{aligned} \tag{4.12}$$

Here, the discrete cost functional $\bar{J}(\mathbf{q}_{hk}, \mathbf{u}_{hk})$ is defined analogously as in Section 3.4 with $\mathbf{u}_{hk} = ((u_{hk}^j)_{j=0}^{M-1})$ and $\mathbf{q}_{hk} = ((q_{hk}^j)_{j=0}^{M-1})$. The additional continuity conditions occurring in the shooting framework have to be also given in a discrete version:

$$\begin{aligned} (s_h^0 - u_0, \phi_h) &= 0 \quad \forall\, \phi_h \in V_h^s, \\ (s_h^{j+1} - u_{hk}^j(\tau_{j+1}), \phi_h) &= 0 \quad \forall\, \phi_h \in V_h^s \ \ \forall\, j \in \{0, \dots, M-1\}. \end{aligned} \tag{4.13}$$

This fully discrete problem is to be compared to its continuous counterpart (3.43)–(3.44).

4.2 Iterative solvers for linear equations

All solution algorithms for complex problems such as the OCP (3.3)–(3.4) or the extended OCP (3.43)–(3.44) consist of basic steps, among the most important of which is solving linear and nonlinear equations or systems.
As an example, Algorithms 2.2 for IMS and 2.3 for DMS are revisited. Step 6 in Algorithm 2.2 (and, correspondingly, Step 3 in Algorithm 2.3) consists of solving certain IVP, which amounts to solving a linear or nonlinear equation in each time step. Analogously, IMS Step 7 (resp. DMS Step 4) requires the solution of variational equations, which means solving a linear equation system per time step. The third part is IMS Step 8 (DMS Step 6), the solution of the respective shooting system, which is again a linear system. The multiple shooting algorithms for parabolic OCP in Chapter 5 consist of almost the same basic steps; there, applications of all methods discussed in the following are given.
This section treats solvers that are commonly used for linear equation systems of the form

$$Ax = b, \quad \text{where } A \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n. \tag{4.14}$$

In the PDE context of Chapter 5, fine discretizations of parabolic OCP lead to so-called large-scale optimization problems, which prohibits the use of direct methods and necessitates iterative linear solvers. In the following, the focus is on methods that can actually be used in the implementation of PDE governed OCP. Important classes of iterative linear solvers are splitting-based ones (like Jacobi, Gauss-Seidel or SSOR) and Krylov subspace methods (e. g., minimum residual methods and projection methods). We concentrate on two specific Krylov subspace methods, one for symmetric positive definite and one for general regular linear systems, whereas splitting-based methods are addressed in the context of preconditioning. The Krylov subspace of dimension k, generated by r^0, is denoted by the symbol $K_k(r^0, A)$. It holds

$$K_k(r^0, A) = \text{span}\{r^0, Ar^0, \cdots, A^{k-1}r^0\}.$$

Detailed derivations of the presented solvers can be found in the literature (see, e. g., the textbooks by Kanzow [62], Meister [87] or Saad [100]).

The conjugate gradient (CG) method. The first example for an iterative solver, the conjugate gradient (CG) algorithm, can be interpreted both as a minimum residual method and as a Galerkin projection method; details on these topics are presented in Kanzow [62]. It is motivated by the following consideration: For a symmetric and positive definite (spd) matrix $A \in \mathbb{R}^n$ and a vector $b \in \mathbb{R}^n$, the solution of (4.14) is equivalent to solving the minimization problem

$$\min_{x\in\mathbb{R}^n} \; f(x) = \frac{1}{2}(Ax, x)_2 + (b, x)_2. \tag{4.15}$$

An spd matrix defines a scalar product $(\cdot,\cdot)_A := (A\cdot,\cdot)_2$; vectors $\{r^i\}_{i=0}^{n-1}$ are called A-orthogonal if $(r^i, r^j)_A = 0$ for all $i, j = 0, \cdots, n-1$ with $i \neq j$. In order to obtain an A-orthogonal basis of $\mathbb{R}^n$, the Gram-Schmidt algorithm can be used.

Lemma 4.1. *Assume A and b as above and x^0 as a starting vector. Let $\{r^i\}_{i=0}^{n-1}$ be an A-orthogonal basis. Then the sequence*

$$x^{k+1} = x^k + \lambda_k r^k,$$

where λ_k solves the one-dimensional minimization problem in direction r^k,

$$f(x^k + \lambda_k r^k) = \min_{\lambda \in \mathbb{R}} f(x^k + \lambda r^k),$$

yields a minimum of $f(x)$ in (4.15) after at most n iterations.

Using basic linear algebra, from this lemma the following CG algorithm 4.1 is obtained that has been developed by Hestenes & Stiefel [55]. As a by-product, the A-orthogonal basis is constructed during the computation and does not have to be specified in advance. This method is used later on to solve the linear systems (or, in the nonlinear case, the linearized Newton equation) that arise in each time step of the intervalwise BVP or IVP (see Chapter 5, Algorithms 5.1 (Step 4) and 5.4 (Step 4)). A variant for large-scale optimization problems required in the IMS context is discussed in the next subsection.

Algorithm 4.1 Conjugate gradient (CG) method by Hestenes & Stiefel [55]

Require: Starting value $x^0 \in \mathbb{R}^n$.
1: Set $p^0 = b - Ax^0, r^0 = p^0$ and $\alpha_0 = \|r^0\|_2^2$; choose tolerance TOL.
2: **for** $k = 0, \cdots, n-1$ **do**
3: **if** $\alpha_k <$ TOL **then**
4: STOP.
5: **else**
6: $v^k = Ap^k, \lambda_k = \frac{\alpha_k}{(v^k, p^k)_2}$
7: $x^{k+1} = x^k + \lambda_k p^k$
8: $r^{k+1} = r^k - \lambda_k v^k$
9: $\alpha_{k+1} = \|r^{k+1}\|_2^2$
10: $p^{k+1} = r^{k+1} + \frac{\alpha_{k+1}}{\alpha_k} p^k$
11: **end if**
12: **end for**

Remark 4.6. Even if this algorithm theoretically produces the exact solution x^* of (4.14) after at most n iterations (see Lemma 4.1) and can thus be viewed as a direct solution method, it is usually counted among the iterative linear solvers for several reasons:

(i) accumulated round-off errors perturb the A-orthogonality of $\{r^i\}_{i=0}^{n-1}$,

(ii) for a large amount of iterations (i. e., a high-dimensional linear system) usually a good approximation to x^* is already found after far less than n iterations,

(iii) large linear systems often result from discretizations; if, with the iterative linear solver, one reaches the level of the discretization error, further iterations do not yield further progress, thus one is not interested in solving the linear system exactly.

The generalized minimum residual (GMRES) method. This second iterative solver can also be derived as a minimum residual scheme, i. e., it is based on the idea of minimizing the Euclidean norm $\|b - Ax\|_2$ of the residual of (4.14). As it is designed for general regular linear systems, the belonging matrix does not define a scalar product, which complicates the proceeding. For a thorough and comprehensible presentation, we refer to Meister [87]. Here, the main parts of the GMRES Algorithm 4.2 are discussed. The first part of

Algorithm 4.2 Generalized minimum residual (GMRES) method by Saad & Schultz [101]

Require: Starting value $x^0 \in \mathbb{R}^n$.

1: Set $r^0 = b - Ax^0, \beta = \|r^0\|_2, v^1 = \frac{r_0}{\beta}, z_1 = \beta$ and $k = 1$, choose tolerance TOL.
2: **while** $\frac{|z_k|}{\beta} >$ TOL **do**
3: $w^k = Av^k$
4: **for** $l = 1 : 1 : k$ **do**
5: $h_{lk} = (v^l, w^k)_2$
6: $w^k = w^k - h_{lk}v^l$
7: **end for**
8: $h_{k+1,k} = \|w^k\|_2$
9: $v^{k+1} = \frac{w^k}{h_{k+1,k}}$
10: **for** $l = 1 : 1 : k - 1$ **do**
11: $h_{lk} = c_l h_{lk} + s_l h_{l+1,k}$
12: $h_{l+1,k} = -s_l h_{lk} + c_l h_{l+1,k}$
13: **end for**
14: $\tau = |h_{kk}| + |h_{k+1,k}|$
15: $\nu = \tau\sqrt{(\frac{h_{kk}}{\tau})^2 + (\frac{h_{k+1,k}}{\tau})^2}$
16: $c_k = \frac{h_{kk}}{\nu}, s_k = \frac{h_{k+1,k}}{\nu}$
17: $h_{kk} = \nu;\ \ h_{k+1,k} = 0$
18: $z_{k+1} = -s_k z_k;\ \ z_k = c_k z_k$
19: $k \leftarrow k + 1$
20: **end while**
21: $y_k = \frac{z_k}{h_{kk}}$
22: **for** $l = k - 1 : -1 : 1$ **do**
23: $y_l = \frac{z_l - \sum_{j=l+1}^{k} h_{lj} y_j}{h_{ll}}$
24: **end for**
25: $x^k = x^0 + \sum_{l=1}^{k} y_l v^l$

the algorithm comprises the orthogonalization process which yields an orthogonal basis $\{v^k\}_{k=1}^n$; this is achieved by a Gram-Schmidt-like Arnoldi algorithm in Steps 3–9. It can be shown that the matrix H_k, consisting of the numbers $\{h_{ij}\}_{i,j=1}^k$, which may be expressed in short by

$$H_k = V_k^\top A V_k,$$

is in fact an upper Hessenberg matrix (here, the columns of the matrix V_k are the basis vectors v^k). The second part of the GMRES algorithm (Steps 10–19) aims at reducing the

original minimum residual problem

$$\min_x \|b - Ax\|_2$$

to a simpler and lower-dimensional Gaussian equalization problem. The essential step is a transformation of the matrix H_k, which is most easily done by Givens rotations (Steps 11–12; alternatively, Householder reflections could be employed). In Steps 21–24, the coefficients y_k are determined which are necessary to express the approximation x^k as a linear combination of the orthogonal basis vectors v^k of the current Krylov subspace $K_k(r^0, A)$ (Step 25).
The GMRES method is employed in Chapter 5 to solve the system of shooting conditions in both IMS and DMS (see Algorithms 5.3 (Step 5) and 5.5 (Step 5)). However, as it is common in practical realizations to save memory, a restarted GMRES method is used, which means that after a fixed number, say m, of iterations, the iterate x^m is computed and taken as a starting vector for a new run of the GMRES algorithm.

Preconditioning. As already indicated above, the convergence behavior of the CG method is well examined. The following result illustrates the necessity of preconditioning techniques. Examples are discussed below.

Theorem 4.2. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric and positive definite matrix, and let $\{x^k\}_{k\in\mathbb{N}_0}$ be the sequence of approximate solutions to* (4.14) *generated by the CG method (see Algorithm 4.1). It holds:*

$$\|x^k - x^*\|_A \leq 2\left(\frac{\sqrt{\operatorname{cond}_2(A)} - 1}{\sqrt{\operatorname{cond}_2(A)} - 1}\right)^k \|x^0 - x^*\|_A. \tag{4.16}$$

Here, x^* is the exact solution of (4.14)*, and* $\operatorname{cond}_2(A)$ *denotes the spectral condition of the matrix A, defined as* $\operatorname{cond}_2(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$.

Proof. The proof uses Tchebychev polynomials and can be found in Meister [87]. □

The relation (4.16) implies that the CG method has a superlinear convergence behavior, as the quotient $(\sqrt{\operatorname{cond}_2(A)} - 1)/(\sqrt{\operatorname{cond}_2(A)} - 1)$ is always smaller than 1 and thus the sequence $\{c_k\}_{k\in\mathbb{N}}$ with

$$c_k = 2\left(\frac{\sqrt{\operatorname{cond}_2(A)} - 1}{\sqrt{\operatorname{cond}_2(A)} - 1}\right)^k$$

converges to zero. However, this convergence may become arbitrarily slow if the condition number $\operatorname{cond}_2(A)$ is large enough. Thus, for ill-conditioned problems, the CG method is inefficient, and it is advisable to use preconditioning techniques.

Remark 4.7. For the GMRES method, there are similar convergence results as the one presented in Theorem 4.2 for the CG algorithm for a symmetric and positive definite system matrix (see Meister [87]). Therefore, the following discussion of preconditioners holds also for GMRES, except for the type of preconditioners to be chosen.

The concept of preconditioning relies on modifying the system matrix A in such a way that the condition number $\mathrm{cond}(\tilde{A})$ of the modified system matrix $\tilde{A}$ is of moderate size. It holds

$$\tilde{A} = P^{-1}A \text{ or, to preserve symmetry, } \tilde{A} = P^{-1}AP^{-\top},$$

where in the latter case the solution variable becomes $y = P^{\top}x$, a linear system which has to be solved in addition. From the above representations of $\tilde{A}$, two aspects should be considered: first, the preconditioning matrix P should be chosen so that $\tilde{A}$ is close to the identity (in order to guarantee an optimal reduction of the condition number). Second, the additional linear system should be easily solvable. These objectives are, however, antagonizing, which can be seen from the two extreme choices $P = I$ (which renders the additional linear system trivial) and $P = A$ (which is the optimal choice for reducing the condition number). In practice, all preconditioners yield a trade-off between these two objectives.
As this work does not focus on the development of good preconditioners, further discussions are skipped; see Meister [87] for details. There are several classes of preconditioners, e. g., the splitting-based ones like Jacobi, Gauss-Seidel or SSOR preconditioners. Other standard preconditioners use incomplete Cholesky (IC) or LU decompositions (ILU) of the system matrix. It is important that, for symmetric methods such as CG, the preconditioning matrix P preserves symmetry and the method can still be applied to the modified system with matrix $\tilde{A}$.
The last paragraph of Subsection 5.1.2 contains a detailed discussion of a symmetric Gauss-Seidel preconditioner for solving the IMS shooting system with a Newton-GMRES method.

4.3 Solvers for nonlinear equations

Important variants of Newton's method. When, in Section 3.4, the extended optimal control formulation was discussed from which the different multiple shooting methods for PDE optimization are derived in Chapter 5, we already hinted at the classical Newton's method as the typical solver for nonlinear equations. The following Algorithm 4.3 for the solution of the general nonlinear equation

$$F(x) = 0 \quad \text{with } F : \mathbb{R}^n \to \mathbb{R}^n \tag{4.17}$$

recalls the two-step formulation (3.49) which was presented for solving the system (3.50). As we always deal with finite-dimensional problems in practice (e. g., systems arising from discretized PDE), the method is formulated in finite dimensions. Note, however, that it can be formulated and analyzed on a function space level as well. Despite its good theoretical properties (local second-order convergence under certain regularization assumptions) Newton's method is rarely applied in this basic form. Nevertheless, there are lots of variants which make it an essential tool for solving nonlinear equations. There are simple variants like the damped Newton method which does not use the full step for the update (i. e., Step 4 in Algorithm 4.3 is replaced by $x^{k+1} = x^k + \lambda\delta x$ for a factor $\lambda \in (0, 1)$).

Algorithm 4.3 Newton's method as a defect correction iterative solver.

Require: Starting value $x^0 \in \mathbb{R}^n$.
1: Set counter $k = 0$, choose tolerance TOL.
2: **while** $\|F(x^k)\| > \text{TOL}$ **do**
3: Solve the system $\nabla F(x^k)\delta x = -F(x^k)$.
4: Compute $x^{k+1} = x^k + \delta x$.
5: $k \leftarrow k + 1$
6: **end while**

But there are also sophisticated concepts like inexact Newton methods; the latter comprise again several subclasses, e. g., quasi-Newton methods (which are neglected in this thesis but remain important in the context of SQP algorithms). We concentrate on a type of inexact Newton methods where Step 3 of Algorithm 4.3 is solved by an iterative method, e. g., one of the solvers mentioned in Section 4.2.

Krylov-Newton methods and globalization techniques. Algorithm 4.4 describes the solution of the Newton system by an iterative solver. For the latter, one of the Krylov subspace algorithms (CG resp. GMRES) presented in the last section is chosen. They are referred to as Krylov-Newton methods.

Algorithm 4.4 Variant of Newton's method with iterative linear solver.

Require: Starting value $x^0 \in \mathbb{R}^n$.
1: Set counter $k = 0$, choose tolerance TOL_1.
2: **while** $\|F(x^k)\| > \text{TOL}_1$ **do**
3: Set counter $i = 0$, choose starting vector δx^0 and tolerance TOL_2.
4: **while** $\|\nabla F(x^k)\delta x^i + F(x^k)\| > \text{TOL}_2$ **do**
5: Carry out one step of the linear iterative solver.
6: $i \leftarrow i + 1$
7: **end while**
8: Compute $x^{k+1} = x^k + \delta x^{\text{end}}$.
9: $k \leftarrow k + 1$
10: **end while**

A Krylov-Newton method that is used for solving the reduced optimal control problems on the shooting subintervals is Steihaug's classical modification of the CG algorithm. It combines the CG method with a trust region globalization technique and is designed for unconstrained OCP of the form

$$\min_q \; j(q)$$

that lead to large-scale optimization problems. The first order necessary optimality condition can be assembled from directional derivatives such as

$$j'(q)(\tau q) = 0.$$

As usual, this root-finding problem may be treated by Newton's method. In each iteration, a linear system of of the type

$$j''(q)(\delta q, \tau q) = -j'(q)(\tau q) \tag{4.18}$$

has to be solved. For many optimization methods relying on quadratic approximations (like the class of sequential quadratic programming (SQP) algorithms) it is an important observation that (4.18) can also be considered as the first order optimality condition of the quadratic problem

$$\min_{\delta q} \; m(\delta q) = j(q) + j'(q)(\delta q) + \frac{1}{2} j''(q)(\delta q, \delta q). \tag{4.19}$$

The reduced gradient and Hessian can be assembled from directional derivatives, and from (4.18) the linear system

$$\nabla^2 j(q)\delta q = -\nabla j(q) \tag{4.20}$$

arises, which is a concrete instantiation of (4.14) with $A = \nabla^2 j(q), x = \delta q$ and $b = -\nabla j(q)$. Now an additional constraint, a so-called trust region radius μ, is added to the quadratic approximation problem (4.19), leading to the constrained problem

$$\min_{\delta q} \; m(\delta q) \quad \text{s. t. } \|\delta q\| \leq \mu. \tag{4.21}$$

In the following this problem is solved by Steihaug's modified CG method (see Algorithm 4.5). As the original problem (4.20) constitutes a Newton equation, this algorithm is an example for a Krylov-Newton method. A similar Newton-GMRES method is used in the algorithms of Chapter 5. A distinguishing feature of Algorithm 4.5 is the presence of three different stopping mechanisms. A minimizer of the quadratic subproblem (4.19) is sought on a circular set with radius μ_j (the index j already hints at the iterative character of the method, i. e., also the trust region radius will be updated during the process). The algorithm terminates if

(i) the quantity γ_k, describing the curvature of the direction g^k, becomes negative. This case represents an extension of the original CG algorithm for matrices A that are not necessarily positive definite; if this happens, we either move to the boundary (Steps 6–7) or accept the previous iterate (Step 10).

(ii) the norm $\|p^{k+1}\|$ exceeds the current trust region radius, because in this case the new iterate lies outside the trust region; if this happens, we move along the direction of p^{k+1} until trust region boundary is reached (Steps 17–18).

(iii) the original Newton step is approximated well enough; in this case, the current iterate is accepted (Step 23).

Furthermore, a comparison of Algorithms 4.1 and 4.5 shows that, with a modified nomenclature, the steps are equivalent in both variants (e. g., Step 8 of Algorithm 4.1 corresponds to Step 21 of Algorithm 4.5, and Step 10 corresponds to Steps 26–27).

Algorithm 4.5 The modified conjugate gradient method by Steihaug [106].

Set $p^0 = 0, r^0 = b, g^0 = r^0$ and counter $k = 0$, choose tolerance TOL.
loop
  Set $\gamma_k = (Ag^k, g^k)$.
  if $\gamma_k \leq 0$ **then**
    if $\mu_j < \infty$ **then**
      Compute $\tau > 0$ so that $\|p^k + \tau g^k\| = \mu_j$.
      Set $x = p^k + \tau g^k$.
      break (*negative curvature*)
    else
      Set $x = p^{k-1}$ (resp. $x = p^0$ if $k = 0$).
      break (*negative curvature*)
    end if
  end if
  Compute $\alpha = \frac{\|r^k\|^2}{\gamma_k}$.
  Set $p^{k+1} = p^k + \alpha g^k$.
  if $\|p^{k+1}\| > \mu_j$ **then**
    Compute $\tau > 0$ so that $\|p^k + \tau g^k\| = \mu_j$.
    Set $x = p^k + \tau g^k$.
    break (*norm of approximation too large*)
  end if
  Compute $r^{k+1} = r^k - \alpha A g^k$.
  if $\frac{\|r^{k+1}\|}{\|r^0\|} <$ TOL **then**
    Set $x = p^{k+1}$.
    break (*approximation sufficiently good*)
  end if
  Compute $\beta = \frac{\|r^{k+1}\|^2}{\|r^k\|^2}$.
  Set $g^{k+1} = r^{k+1} + \beta g^k$.
  $k \leftarrow k + 1$
end loop

Matrix-free computation. The presented algorithms require the system matrix A and the righthand side b as input and yield the solution x or at least an approximation $\tilde{x}$ of predetermined quality as output. However, in large-scale optimization, it is often impossible to assemble the system matrix due to the numerical costs.
In this last paragraph of Chapter 4 the concept of matrix-free computation is discussed in an abstract manner and for the simplest example. Concrete possibilities for avoiding explicit matrix assemblation are encountered in Chapter 5 (see, e. g., equations (5.13) or (5.24)). Here, the following case of a two-dimensional system is considered:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \tag{4.22}$$

This system can be transformed into

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1, \\ a_{21}x_1 + a_{22}x_2 &= b_2. \end{aligned}$$

Now, with $y_1 = a_{11}x_1 + a_{12}x_2$ and $y_2 = a_{21}x_1 + a_{22}x_2$, the resulting vector y is called a matrix-vector product. Assume that a CG or GMRES method is employed to solve (4.22). If y_1 and y_2 can be expressed, e. g., by solving certain additional equations (which will be the case in Chapter 5), then all the expressions Ap^k, Av^k and Ag^k in Algorithms 4.1, 4.2 and 4.5 can be replaced by y^k, which is then iteratively updated by the respective method (CG or GMRES).

5 Multiple Shooting Approaches for PDE Constrained OCP

The current chapter deals with several multiple shooting variants for parabolic OCP. The presentation of ODE optimal control problems and their numerical solution by means of shooting methods in Section 2.3 introduced two different formulations of the multiple shooting method. These two approaches reflect the more general distinction between direct and indirect methods for OCP (see Rao [96] for an overview). Therefore, they are called direct multiple shooting (DMS) and indirect multiple shooting (IMS), respectively. This chapter describes both formulations in detail for PDE governed OCP. Hesse [52] stated an equivalence of DMS and IMS in an abstract setting (compare also Hesse & Rannacher [54]) which is not obvious from the algorithms presented in Section 2.3. Hesse's DMS formulation consists of the same components as IMS, but it deviates significantly from the well-established DMS solution routine presented in Section 2.3 in the ODE case. We provide a comprehensible presentation of the interrelations between the shooting variants and formulations. Sections 5.1 and 5.2 derive IMS and DMS from the same underlying problem. The discussion reveals the similar structure of both approaches but, in their algorithmic realization, we recognize important differences. In Section 5.3, the DMS formulation from Section 5.2 and the one commonly used in ODE optimal control (cf. Subsection 2.3.2) are proven to be identical. The latter one is interpreted as a reduced formulation of the former one. Further aspects of both shooting variants are presented in Section 5.4. By means of numerical tests, the theoretical results are substantiated in the concluding Section 5.5. Central parts of this chapter have already been submitted as a contribution to an anthology in Carraro & Geiger [21].

5.1 Indirect multiple shooting (IMS)

The first variant of multiple shooting, the IMS method, has not been employed much in the PDE optimal control framework. Unlike its more widely used direct counterpart, it does not reduce the OCP boundary value problem to a series of initial value problems, but splits the original BVP (see Remark 3.8) into a sequence of identical problems on the shooting subintervals (3.42). IMS is introduced by Hesse [52] and first published in Hesse & Kanschat [53], where the authors' focus lies on the application of error estimators in the multiple shooting context. A thorough presentation of the method itself is omitted. Therefore, we develop the main ideas and the overall structure of IMS in Subsection 5.1.1. Subsequently, the implementational details of the IMS method are presented in Subsection 5.1.2, where concrete algorithms for several subproblems are proposed.

5.1.1 Structure of IMS

In order to create a suitable framework for multiple shooting methods as solvers for parabolic OCP of type (3.3)–(3.4), the problem has to be modified by including additional continuity conditions. The first order optimality conditions of the resulting extended OCP (3.43)–(3.44) lead to the formal Newton system (3.50). Recall that this high-dimensional linear system (respectively, its discrete counterpart) is neither assembled nor solved in practice but serves for illustratory purposes. The derivation of IMS is based on a splitting of (3.50) into two parts; the current subsection describes this splitting, which provides IMS with the structure of a two-step fixed-point iteration.
Remember that in system (3.50), the abbreviatory matrix and vector entries have to be interpreted componentwise, e. g., $\bar{\mathcal{L}}'_u$ stands for $(\bar{\mathcal{L}}'_{u^0}, \cdots, \bar{\mathcal{L}}'_{u^{M-1}})^\top$. It is important that all variables $\delta u^j, \delta z^j, \delta q^j, \delta s^j$ and $\delta\lambda^j$ are independent. They are regrouped according to the following scheme:

$$\left(\begin{array}{ccc|cc} 0 & \bar{\mathcal{L}}''_{uz} & \bar{\mathcal{L}}''_{qz} & \bar{\mathcal{L}}''_{sz} & 0 \\ \bar{\mathcal{L}}''_{zu} & \bar{\mathcal{L}}''_{uu} & 0 & 0 & \bar{\mathcal{L}}''_{\lambda u} \\ \bar{\mathcal{L}}''_{zq} & 0 & \bar{\mathcal{L}}''_{qq} & 0 & 0 \\ \hline \bar{\mathcal{L}}''_{zs} & 0 & 0 & 0 & \bar{\mathcal{L}}''_{\lambda s} \\ 0 & \bar{\mathcal{L}}''_{u\lambda} & 0 & \bar{\mathcal{L}}''_{s\lambda} & 0 \end{array}\right) \left(\begin{array}{c} \delta z \\ \delta u \\ \delta q \\ \hline \delta s \\ \delta\lambda \end{array}\right) = - \left(\begin{array}{c} \bar{\mathcal{L}}'_z \\ \bar{\mathcal{L}}'_u \\ \bar{\mathcal{L}}'_q \\ \hline \bar{\mathcal{L}}'_s \\ \bar{\mathcal{L}}'_\lambda \end{array}\right). \tag{5.1}$$

Now we deal with two subsets of variables. On the one hand, $\delta u^j, \delta z^j, \delta q^j$, which are the intervalwise counterparts of the corresponding update variables of the global problem. On the other hand, $\delta s^j, \delta\lambda^j$, which comprise all artificially introduced additional variables of the extended OCP (3.43)–(3.44). Successive solution of the corresponding two subsystems of (5.1) introduces inherent dependencies between the variables. In a first solution step, $\mathbf{s} = (s^j)_{j=0}^M$ and $\boldsymbol{\lambda} = (\lambda^j)_{j=0}^M$ are fixed to solve the intervalwise boundary value problems

$$(\!(\partial_t u^j, \delta z)\!) + a(u^j)(\delta z) + b(q^j)(\delta z) - (\!(f|_{I_j}, \delta z)\!) + (u^j(\tau_j) - s^j, \delta z(\tau_j)) = 0, \tag{5.2a}$$
$$J_u^{j\prime}(q^j, u^j)(\delta u) - (\!(\partial_t z^j, \delta u)\!) + a'_u(u^j)(\delta u, z^j) + (z^j(\tau_{j+1}) - \lambda^{j+1}, \delta u(\tau_{j+1})) = 0, \tag{5.2b}$$
$$J_q^{j\prime}(q^j, u^j)(\delta q) + b'_q(q^j)(\delta q, z^j) = 0. \tag{5.2c}$$

Remark 5.1. As there is no need to fit given values at the end time τ_M for the global state variable and at the initial time τ_0 for the global adjoint variable, the corresponding variables s^M and λ^0 are redundant. They are skipped to decrease the size of the shooting system. Furthermore, the variables s^0 and λ^M could be replaced by the known initial values $s^0 \equiv u_0$ and $\lambda^M \equiv 0$, or $\lambda^M \equiv \hat{u}_T$ in case of an end-time functional. The main reason for keeping them in the system is the resulting simplification in the implementation of the method. By this, all shooting intervals can be treated in an identical manner.

The equations (5.2) correspond to $\bar{\mathcal{L}}'_{z^j}(\xi)(\delta z) = 0, \bar{\mathcal{L}}'_{u^j}(\xi)(\delta u) = 0$ and $\bar{\mathcal{L}}'_{q^j}(\xi)(\delta q) = 0$ in (3.48). The variables u^j, z^j and q^j now depend on s^j and λ^{j+1}. The BVP character of

the intervalwise KKT systems results from the forward-backward structure and the full coupling of the state and adjoint equations. The parameter s^j is the initial value for u^j at τ_j, and λ^{j+1} is the initial value for z^j at the subinterval endpoint τ_{j+1}. The states $u^j(s^j, \lambda^{j+1})$ and $z^j(s^j, \lambda^{j+1})$ are coupled via the control equation (5.2c). Note that, in the ODE case in Subsection 2.3.1, the control equation has been solved for q^j and the result was inserted into the state equation. This left us with a two-component BVP in u^j and z^j, and the control was computed performing a simple update step. In Remark 2.8, it was claimed that this reduction is not possible in general, which is why the three-component BVP (5.2) is left unaltered in the PDE case.

Remark 5.2. The initial choice of the parameters s^j and λ^{j+1} is essential, as its quality influences the convergence of Newton's method. For ODE problems, there exist several suggestions; e. g., additional information on the solution, if available, could improve the initial guesses (for an example, see Bulirsch & Stoer [20]), or alternatively, one could employ homotopy methods as in Lory [77]. This question has potential for further research but is not considered here.

Together the three equations (5.2) bear the same structure on each shooting subinterval as the global KKT system (3.32). However, as the intervalwise solutions do not fit together at the subinterval endpoints, the solution is globally discontinuous due to the artificially chosen initial values s^j and λ^{j+1}. As pointed out in Section 3.4, this contradicts the embedding $X \hookrightarrow C(\bar{I}; H)$. Therefore, the local solutions $u^j(s^j, \lambda^{j+1}), z^j(s^j, \lambda^{j+1})$ and $q^j(s^j, \lambda^{j+1})$ are used to update s^j and λ^{j+1} in the second solution step for (5.1), which consists of solving the following system (corresponding to $\bar{\mathcal{L}}'_{\lambda^j}(\xi)(\delta\lambda) = 0, \bar{\mathcal{L}}'_{s^j}(\xi)(\delta s) = 0$):

$$\begin{aligned} (s^0 - u_0, \delta\lambda) &= 0, \\ \begin{bmatrix} (\lambda^j - z^j(\tau_j; s^j, \lambda^{j+1}), \delta s) = 0 \\ (s^j - u^{j-1}(\tau_j; s^{j-1}, \lambda^j), \delta\lambda) = 0 \end{bmatrix}, &\quad (j = 1, \ldots, M-1), \\ (\lambda^M, \delta s) &= 0. \end{aligned} \tag{5.3}$$

These equations constitute the shooting system, which is the part of (3.48) actually solved by Newton's method. Abbreviating the above shooting equations (5.3) by $F(\mathbf{s}, \boldsymbol{\lambda}) = 0$, we have to solve

$$\nabla_{(\mathbf{s},\boldsymbol{\lambda})} F(\mathbf{s}, \boldsymbol{\lambda}) \begin{pmatrix} \delta\mathbf{s} \\ \delta\boldsymbol{\lambda} \end{pmatrix} = -F(\mathbf{s}, \boldsymbol{\lambda}) \tag{5.4}$$

instead of the whole system (3.50). This results in improved values

$$\mathbf{s}^{\text{new}} = \mathbf{s} + \delta\mathbf{s}, \ \boldsymbol{\lambda}^{\text{new}} = \boldsymbol{\lambda} + \delta\boldsymbol{\lambda},$$

with which step one described above is restarted. The asserted structure of a two-step fixed-point iteration, alternating between computing (u^j, z^j, q^j) and updating (s^j, λ^j), is evident. The whole process is resumed in the following Algorithm 5.1.

Algorithm 5.1 Indirect multiple shooting for PDE governed OCP

Require: Decomposition $I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1}(\tau_j, \tau_{j+1}]$, initial values $\{(s_0^j, \lambda_0^{j+1})_{j=0}^{M-1}\}$.
1: Set $k = 1$.
2: **while** Shooting conditions (5.3) not fulfilled **do**
3: **for** $j = 0$ to $M - 1$ **do**
4: Solve intervalwise boundary value problems (5.2).
5: **end for**
6: Solve (5.4), compute update $\{(s_k^j, \lambda_k^{j+1})_{j=0}^{M-1}\}$ of initial values, set $k \leftarrow k + 1$.
7: **end while**

5.1.2 Algorithmic description

We now focus on steps 4 and 6 of Algorithm 5.1, namely the solution of the intervalwise BVP (5.2) (the first part of our two stage fixed-point problem) and the solution of the shooting system (5.4) (the second part, correspondingly).

The intervalwise OCP. As the intervalwise BVP (5.2) are smaller copies of the corresponding global problem (3.48), strategies that have been proved and tested in the global context are now applied to the intervalwise problems. Furthermore, these problems can be solved independently which allows for parallelization of the multiple shooting code. There are two main approaches to solving a problem such as (5.2). The all-at-once approach solves (5.2) as a whole, treating u^j, z^j and q^j as independent variables. Well-known variants are, e. g., the Schur complement method (for details, see Choi et al. [23] and the references therein) and the nullspace method which has, e. g., been described by Vicente [113]. Note that for all-at-once strategies, one usually has to find suitable preconditioners, as the saddle point problems resulting from the KKT conditions are usually ill-conditioned. Keller, Gould & Wathen [63] suggested a preconditioner which works well within a GMRES solver environment.
Instead of going into the details of all-at-once solution strategies, a reduced approach is presented which has already been sketched in Section 3.3 for the global OCP and has also been implemented for our numerical examples. Its important feature is the reduction of the set of independent variables from (q^j, u^j) to the control q^j. Hence, the interval state $u^j = u^j(q^j)$ is interpreted in terms of the interval control by means of a local solution operator $S_j : Q^j \to X^j$. This procedure is described in a similar way in Meidner & Vexler [85], and a detailed presentation can also be found in the textbook Hinze et al. [59]. To clarify the notation used in Algorithm 5.2 below, the reduced cost functional on subinterval I_j is defined as

$$\hat{J}(q^j) := J^j(q^j, u^j(q^j)). \tag{5.5}$$

Following the proceeding of Section 3.3, we find expressions for the intervalwise reduced first order and second order directional derivatives. These are given by $\hat{J}_q'(q^j)(\delta q)$ and $\hat{J}_{qq}''(q^j)(\delta q_2, \delta q_1)$, which should be compared to (3.34) and (3.35), respectively. After solving the intervalwise state and adjoint equations (5.2a) and (5.2b) (for the latter, local solution operators $T_j : Q^j \to X^j$ are introduced), we obtain in analogy to (3.34) and the discussion

thereafter

$$\hat{J}'_q(q^j)(\delta q) = \bar{\mathcal{L}}'_q(\xi)(\delta q) = \alpha((q^j, \delta q)) + b'_q(q^j)(\delta q, z^j). \tag{5.6}$$

For given δq^j, the intervalwise tangent equation (cf. (3.39))

$$((\partial_t \delta u^j, \varphi)) + a'_u(u^j)(\delta u^j, \varphi) + b'_q(q^j)(\delta q^j, \varphi) + (\delta u^j(\tau_j), \varphi(\tau_j)) = 0 \tag{5.7}$$

is solved for δu^j. Then the intervalwise extra adjoint equation (cf. (3.40))

$$\begin{aligned} -((\partial_t \delta z^j, \psi)) + a'_u(u^j)(\psi, \delta z^j) + a''_{uu}(u^j)(\delta u^j, \psi, z^j) \\ + ((\delta u^j, \psi)) + (\delta z^j(\tau_{j+1}), \psi(\tau_{j+1})) = 0 \end{aligned} \tag{5.8}$$

is solved for δz^j. These solutions permit the representation (cf. (3.41))

$$\hat{J}''_{qq}(q^j)(\delta q^j, \chi) = b'_q(q^j)(\chi, \delta z^j) + b''_{qq}(q^j)(\delta q^j, \chi, z^j) + \alpha((\delta q^j, \chi)) \tag{5.9}$$

of the second order directional derivative of the reduced objective functional. As agreed in Section 3.4, a distributed functional structure is assumed, i. e., $\kappa_1 = 1, \kappa_2 = 0$. Further modifications are necessary in case of an end-time functional. In order to evaluate the second order directional derivative (5.9), the solution δz^j of the extra adjoint equation is required. To solve the latter, the solution δu^j of the tangent equation has to be known. Thus, to evaluate one second order directional derivative of $\hat{J}(q^j)$, two additional linear equations have to be solved.

In principle, the reduced Hessian $\nabla^2 \hat{J}(q^j)$ could be assembled by determining all second order directional derivatives for a whole basis $\{\delta q^i\}_{i=1}^{\dim Q^j}$ (this is described in detail in Meidner [84]), but this is very expensive for high-dimensional control spaces. In fact, we then have to solve $2 \cdot \dim Q^j$ additional linear problems. It is more efficient to employ a Newton-CG method as described in Section 4.3. This permits a matrix-free computation, and one evaluation of (5.9) replaces a matrix-vector product of the form $\nabla^2 \hat{J}(q^j) \cdot \delta q$. One then has to solve two additional linear problems per CG iteration.

The following algorithm can now replace step 4 in the overall Algorithm 5.1:

In Section 5.3, the concept of reduced control problems is revisited in a more complicated form in the context of different DMS techniques.

The shooting system. The solution of (5.4) by Newton's method involves the Jacobian matrix $\nabla_{(\mathbf{s},\boldsymbol{\lambda})} F(\mathbf{s}, \boldsymbol{\lambda})$ of the shooting conditions (5.3). Despite having substantially reduced the size of the Newton system (in this regard, (5.4) must be compared to (3.50)), the effort for explicitly assembling this Jacobian is still not manageable. Therefore, a matrix-free method is employed to solve (5.4), in our case a Newton-GMRES approach. Algorithm 5.3 below comprises the essential steps.

The explicit form of system (5.4) which underlies the implementation of the examples presented in Section 5.5.1 reads

$$\begin{bmatrix} A & B_0 & & \cdots & 0 \\ C_0 & A & & & \vdots \\ & & \ddots & & \\ \vdots & & & A & B_{M-2} \\ 0 & \cdots & & C_{M-2} & A \end{bmatrix} \begin{bmatrix} \delta y^0 \\ \delta y^1 \\ \vdots \\ \delta y^{M-2} \\ \delta y^{M-1} \end{bmatrix} = - \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_{M-2} \\ F_{M-1} \end{bmatrix}. \tag{5.10}$$

Algorithm 5.2 Solution of the intervalwise BVP (reduced approach)

Require: Set $k = 0$, prescribe tolerance TOL_1 and initial control q_0^j.
1: **while** $\|\nabla \hat{J}(q_k^j)\| > TOL_1$ **do**
2: Solve state equation (5.2a).
3: Solve adjoint equation (5.2b).
4: Compute gradient $\nabla \hat{J}(q_k^j)$ of reduced cost functional.
5: Set $i = 0$, prescribe tolerance TOL_2 and $\delta q_{k,0}^j$.
6: **while** $\|\delta q_{k,i+1}^j - \delta q_{k,i}^j\| > TOL_2$ **do**
7: Compute matrix-vector product $\nabla^2 \hat{J}(q_k^j)\delta q_{k,i}^j$.
8: Solve system $\nabla^2 \hat{J}(q_k^j)\delta q_{k,i}^j = -\nabla \hat{J}(q_k^j)$ by a Newton-CG method (this requires solving the tangent equation (5.7) and extra adjoint equation (5.8) in each iteration; they are obtained by linearization of (5.2a) and (5.2b), respectively).
9: **end while**
10: Set $k \leftarrow k + 1$ and $q_{k+1}^j = q_k^j + \delta q_{k,end}^j$.
11: **end while**

For $j = 0, \ldots, M-1$, the solution subvectors are explicitly given as $\delta y^j := (\delta s^j, \delta \lambda^{j+1})^\top$. Furthermore, the submatrices are

$$A = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad B_j = \begin{bmatrix} 0 & 0 \\ -z_s^{j+1\prime}(\tau_{j+1}) & -z_\lambda^{j+1\prime}(\tau_{j+1}) \end{bmatrix}, \quad C_j = \begin{bmatrix} -u_s^{j\prime}(\tau_{j+1}) & -u_\lambda^{j\prime}(\tau_{j+1}) \\ 0 & 0 \end{bmatrix}$$

for $j = 0, \ldots, M-2$, and the righthand side subvectors are given by

$$\begin{aligned} F_j &= \begin{bmatrix} s^j - u^{j-1}(\tau_j; s^{j-1}, \lambda^j) \\ \lambda^{j+1} - z^{j+1}(\tau_{j+1}; s^{j+1}, \lambda^{j+2}) \end{bmatrix}, \quad (j = 1, \ldots, M-2), \\ F_0 &= \begin{bmatrix} s^0 - u_0 \\ \lambda^1 - z^1(\tau_1; s^1, \lambda^2) \end{bmatrix}, \quad F_{M-1} = \begin{bmatrix} s^{M-1} - u^{M-2}(\tau_{M-1}; s^{M-2}, \lambda^{M-1}) \\ \lambda^M \end{bmatrix}. \end{aligned}$$

The Jacobian in (5.10) has a block tridiagonal structure. The diagonal blocks are identity matrices of twice the size of the spatial dimension, i. e., in the discretized case they have dimension $2R \times 2R$, where $R = \dim V_h^s$ in (4.6). The blocks on the first upper and lower diagonals comprise two $R \times R$ zero submatrices and two matrices of the same size given as derivatives of the intervalwise OCP solutions u^j and z^j with respect to their initial values s^j and λ^{j+1}, respectively. Thus, the whole Jacobian ∇F is of size $2RM \times 2RM$, where M is the number of shooting intervals.
The derivatives $u_s^{j\prime}$ and $z_s^{j\prime}$ are obtained as solutions of the system

$$\begin{aligned} (\!(\partial_t u_s^{j\prime}, \varphi)\!) + a_u'(u^j)(u_s^{j\prime}, \varphi) + b_q'(q^j)(q_s^{j\prime}, \varphi) + (u_s^{j\prime}(\tau_j) - \delta s^j, \varphi(\tau_j)) &= 0 \quad \forall \varphi \in X_j, \\ J_{uu}^{j\prime\prime}(u^j)(u_s^{j\prime}, \psi) - (\!(\partial_t z_s^{j\prime}, \psi)\!) + a_{uu}''(u^j)(u_s^{j\prime}, \psi, z^j) & \\ + a_u'(u^j)(\psi, z_s^{j\prime}) + (z_s^{j\prime}(\tau_{j+1}), \psi(\tau_{j+1})) &= 0 \quad \forall \psi \in X_j, \\ J_{qq}^{j\prime\prime}(q^j)(q_s^{j\prime}, \chi) - b_{qq}''(q^j)(q_s^{j\prime}, \chi, z^j) - b_q'(q^j)(\chi, z_s^{j\prime}) &= 0 \quad \forall \chi \in Q_j. \end{aligned} \tag{5.11}$$

This system is the derivative of the intervalwise optimality conditions (5.2) with respect to s^j in direction δs^j. Analogously, to compute $u_\lambda^{j\prime}$ and $z_\lambda^{j\prime}$, the corresponding system

$$\begin{aligned}
((\partial_t u_\lambda^{j\prime}, \varphi)) + a_u'(u^j)(u_\lambda^{j\prime}, \varphi) + b_q'(q^j)(q_\lambda^{j\prime}, \varphi) + (u_\lambda^{j\prime}(\tau_j), \varphi(\tau_j)) &= 0 \quad \forall \varphi \in X_j, \\
J_{uu}^{j\prime\prime}(u^j)(u_\lambda^{j\prime}, \psi) - ((\partial_t z_\lambda^{j\prime}, \psi)) + a_{uu}''(u^j)(u_\lambda^{j\prime}, \psi, z^j) & \\
+ a_u'(u^j)(\psi, z_\lambda^{j\prime}) + (z_\lambda^{j\prime}(\tau_{j+1}) - \delta\lambda^{j+1}, \psi(\tau_{j+1})) &= 0 \quad \forall \psi \in X_j, \\
J_{qq}^{j\prime\prime}(q^j)(q_\lambda^{j\prime}, \chi) - b_{qq}''(q^j)(q_\lambda^{j\prime}, \chi, z^j) - b_q'(q^j)(\chi, z_\lambda^{j\prime}) &= 0 \quad \forall \chi \in Q_j,
\end{aligned} \tag{5.12}$$

has to be solved which is the derivative of (5.2) with respect to λ^{j+1} in direction $\delta\lambda^{j+1}$. Solving the systems (5.11) and (5.12) yields the directional derivatives $u_s^{j\prime}(t; \delta s^j)$, $z_s^{j\prime}(t; \delta s^j)$, $u_\lambda^{j\prime}(t; \delta\lambda^{j+1})$, and $z_\lambda^{j\prime}(t; \delta\lambda^{j+1})$. In the examples of Section 5.5, both systems are solved by a fixed-point iteration. Now, the matrix-vector product can be written in the following form:

$$\nabla_{(\mathbf{s},\boldsymbol{\lambda})} F(\mathbf{s}, \boldsymbol{\lambda}) \begin{pmatrix} \delta\mathbf{s} \\ \delta\boldsymbol{\lambda} \end{pmatrix} = \begin{bmatrix}
\delta s^0 \\
\delta\lambda^1 - z_s^{1\prime}(\tau_1; \delta s^1) - z_\lambda^{1\prime}(\tau_1; \delta\lambda^2) \\
\delta s^1 - u_s^{0\prime}(\tau_1; \delta s^0) - u_\lambda^{0\prime}(\tau_1; \delta\lambda^1) \\
\delta\lambda^2 - z_s^{2\prime}(\tau_2; \delta s^2) - z_\lambda^{2\prime}(\tau_2; \delta\lambda^3) \\
\vdots \\
\delta s^{M-2} - u_s^{M-3\prime}(\tau_{M-2}; \delta s^{M-3}) - u_\lambda^{M-3\prime}(\tau_{M-2}; \delta\lambda^{M-2}) \\
\delta\lambda^{M-1} - z_s^{M-1\prime}(\tau_{M-1}; \delta s^{M-1}) - z_\lambda^{M-1\prime}(\tau_{M-1}; \delta\lambda^M) \\
\delta s^{M-1} - u_s^{M-2\prime}(\tau_{M-1}; \delta s^{M-2}) - u_\lambda^{M-2\prime}(\tau_{M-1}; \delta\lambda^{M-1}) \\
\delta\lambda^M - J''(u_s^{M-1\prime}(\tau_M; \delta s^{M-1})) - J''(u_\lambda^{M-1\prime}(\tau_M; \delta\lambda^M))
\end{bmatrix}. \tag{5.13}$$

Computation of the whole Jacobian ∇F with the sensitivity method (see, e. g., Hinze et al. [59]) requires for each pair of derivatives $u_s^{j\prime}, z_s^{j\prime}$ and $u_\lambda^{j\prime}, z_\lambda^{j\prime}$ the solution of equations (5.11) or (5.12) for δs^j and $\delta\lambda^{j+1}$. For these directions, a complete set of basis functions of the discrete space V_h^s defined in Subsection 4.1.2 has to be run through. This means that one $2R \times 2R$ block B_j or C_j requires the solution of $2R$ linear boundary value problems, which amounts to a total number of $2R(M-1)$ linear boundary value problems for the whole Jacobian. This is costly on highly refined spatial meshes. To avoid this, (5.4) is solved by a matrix-free approach similar to step 8 of Algorithm 5.2. For the latter, a Newton-CG method has been chosen, which requires the solution of two additional problems (5.7) and (5.8) in each iteration. Similarly, for the solution of (5.4), we employ a Newton-GMRES iterative method. As the matrix in (5.10) is not symmetric, a CG approach is not expected to work for the shooting system. In this framework, equations (5.11) and (5.12) have to be solved in addition once per iteration of the Newton-GMRES method. This approach resembles the adjoint approach for solving reduced optimal control problems (cf. again Hinze et al. [59]). The corresponding Algorithm 5.3 can now substitute step 6 in Algorithm 5.1.

Preconditioned IMS. Example 2.1 suggests that the condition of the Jacobian ∇F deteriorates with an increasing number of shooting intervals. Thus the use of a preconditioner becomes necessary. In Section 4.2 the concept of preconditioning was briefly

Algorithm 5.3 Solution of the IMS shooting system (matrix-free approach)

Require: Shooting variables $(\mathbf{s}_k, \boldsymbol{\lambda}_k)$, intervalwise OCP solutions u^j, z^j

1: Build up residual $-F(\mathbf{s}_k, \boldsymbol{\lambda}_k)$.
2: Set $i = 0$, prescribe tolerance TOL and choose $(\delta\mathbf{s}_k^{(0)}, \delta\boldsymbol{\lambda}_k^{(0)})$.
3: **while** $\|\nabla F(\mathbf{s}_k, \boldsymbol{\lambda}_k)(\delta\mathbf{s}_k^{(i)}, \delta\boldsymbol{\lambda}_k^{(i)}) + F(\mathbf{s}_k, \boldsymbol{\lambda}_k)\| > TOL$ **do**
4: Compute matrix-vector product $\nabla F(\mathbf{s}_k, \boldsymbol{\lambda}_k)(\delta\mathbf{s}_k^{(i)}, \delta\boldsymbol{\lambda}_k^{(i)})$.
5: Solve system $\nabla F(\mathbf{s}_k, \boldsymbol{\lambda}_k)(\delta\mathbf{s}_k^{(i)}, \delta\boldsymbol{\lambda}_k^{(i)}) = -F(\mathbf{s}_k, \boldsymbol{\lambda}_k)$ by a Newton-GMRES type method (this requires the solution of two additional BVP, the linearizations (5.11) and (5.12) of (5.2) w. r. t. $\mathbf{s}$ resp. $\boldsymbol{\lambda}$, in each iteration).
6: **end while**
7: Set $k \leftarrow k + 1$ and $\mathbf{s}_{k+1} = \mathbf{s}_k + \delta\mathbf{s}_k^{end}, \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \delta\boldsymbol{\lambda}_k^{end}$.

discussed, which is now concretized in the context of the Newton-GMRES solver (see Step 5 of Algorithm 5.3) for the IMS continuity conditions (5.3). These conditions lead to the shooting system (5.4), abstractly rewritten as

$$K\delta x = b. \tag{5.14}$$

Our presentation follows Hesse [52] who tested different preconditioners, following the work by Heinkenschloss [50] on preconditioned DMS (cf. the last paragraph of Subsection 5.2.2). The focus lies on a symmetric Gauss-Seidel (SGS) preconditioner which is easily implemented. Its structure is presented, a matrix-free variant is dicussed, and its performance is described. Numerical results are presented in Subsection 5.5.1.
Splitting-based iterative methods rely on an additive decomposition of the given system matrix into a subdiagonal, diagonal and superdiagonal part. In the case of (5.14), the system matrix K resolves into three matrices, denoted from left to right by L, D and U, which contain each only one nonzero block diagonal:

$$K = \begin{bmatrix} 0 & 0 & & \cdots & 0 \\ C_0 & 0 & & & \vdots \\ & & \ddots & & \\ \vdots & & C_{M-1} & 0 & 0 \\ 0 & \cdots & & C_{M-2} & 0 \end{bmatrix} + \begin{bmatrix} A & 0 & & \cdots & 0 \\ 0 & A & & & \vdots \\ & & \ddots & & \\ \vdots & & & A & 0 \\ 0 & \cdots & & 0 & A \end{bmatrix} + \begin{bmatrix} 0 & B_0 & & \cdots & 0 \\ 0 & 0 & B_1 & & \vdots \\ & & \ddots & & \\ \vdots & & & 0 & B_{M-2} \\ 0 & \cdots & & 0 & 0 \end{bmatrix}.$$

The prefactor matrix P for the SGS preconditioner is then given by

$$P = (D + L)D^{-1}(D + U). \tag{5.15}$$

Due to the structure of Krylov subspace iterative algorithms (see, e. g., the GMRES method in Algorithm 4.2), the preconditioner must be applied to the residual $K\delta x - b$ or a similarly structured equation. Then the objective is to compute an expression of the form

$$P^{-1}(K\delta x - b) =: w$$

in each GMRES step. If we denote the residual by r and recall the structure of P, where the matrix D and its inverse D^{-1} are both given by the identity matrix, the resulting system is given by

$$(D+U)^{-1}(D+L)^{-1}r = w.$$

This system is solved in two steps, namely

$$\begin{aligned} (D+L)v &= r, \\ (D+U)w &= v. \end{aligned} \tag{5.16}$$

Recall that the block matrix L essentially comprises derivatives of the intervalwise state solution u^j with respect to s^j and λ^{j+1} (and the entries of the block matrix U are analogously given by derivatives of z^j with respect to s^j and λ^{j+1}). It follows that solving the systems (5.16) in a matrix-free manner is equivalent to the solution of two additional linear boundary value problems of the type (5.11) and (5.12) per shooting interval, but with different righthand sides. To illustrate this, the first system is written explicitly as

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_{M-2} \\ r_{M-1} \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 - C_0 v_0 \\ v_2 - C_1 v_1 \\ \vdots \\ v_{M-2} - C_{M-3} v_{M-3} \\ v_{M-1} - C_{M-2} v_{M-2} \end{bmatrix}$$

where, again exemplarily, the j-th equation (for $j > 0$) corresponds to

$$\begin{aligned} r_j^{(1)} &= \delta s^j - u_s^{j-1}(\tau_j; \delta s^{j-1}) - u_\lambda^{j-1}(\tau_j; \delta\lambda^j), \\ r_j^{(2)} &= \delta\lambda^{j+1}. \end{aligned}$$

As the use of this SGS preconditioner necessitates two additional linear BVP solves per GMRES step, the number of linear BVP of type (5.11) or (5.12) to be solved per GMRES iteration is doubled by the preconditioner. For an efficient preconditioning, the number of GMRES steps must be reduced drastically. In Subsection 5.5.1, several numerical results are presented. As these examples illustrate that the SGS preconditioner is not efficient, for the remainder of our thesis, most computations are carried out without preconditioning. We have not been able to detect the same positive effects of the SGS preconditioner as Hesse [52].

5.2 Direct multiple shooting (DMS)

As mentioned in Section 2.1, the majority of the literature on multiple shooting for PDE governed OCP concentrates on DMS methods (see, e. g., Heinkenschloss [50], Serban et al. [105] or Ulbrich [110]). In the following, a variant of direct shooting is introduced which is based on a similar concept as IMS in the last section. Again, the overall structure of the

DMS approach is discussed first in Subsection 5.2.1 before presenting algorithmic details in Subsection 5.2.2. The introduced DMS method is similar to the IMS approach from Section 5.1, but it deviates substantially from the 'classical' DMS scheme known from ODE optimal control (see also Subsection 2.3.2). The latter has been established by Bock and his co-workers (see, e. g., [11, 13, 15]), and many algorithmic and implementational details can be found in Leineweber [73]. This alternative DMS approach is transferred to the PDE framework in Section 5.3, where it is also shown that the two seemingly different DMS variants are nevertheless equivalent.

5.2.1 Structure of DMS

In contrast to the ODE framework, DMS is derived similarly to IMS by splitting the solution process of system (3.50) into two parts. However, DMS relies on a different regrouping of the variables which is illustrated by the following scheme:

$$\left(\begin{array}{cc|ccc} 0 & \bar{\mathcal{L}}''_{uz} & \bar{\mathcal{L}}''_{qz} & \bar{\mathcal{L}}''_{sz} & 0 \\ \bar{\mathcal{L}}''_{zu} & \bar{\mathcal{L}}''_{uu} & 0 & 0 & \bar{\mathcal{L}}''_{\lambda u} \\ \hline \bar{\mathcal{L}}''_{zq} & 0 & \bar{\mathcal{L}}''_{qq} & 0 & 0 \\ \bar{\mathcal{L}}''_{zs} & 0 & 0 & 0 & \bar{\mathcal{L}}''_{\lambda s} \\ 0 & \bar{\mathcal{L}}''_{u\lambda} & 0 & \bar{\mathcal{L}}''_{s\lambda} & 0 \end{array}\right) \left(\begin{array}{c} \delta z \\ \delta u \\ \hline \delta q \\ \delta s \\ \delta\lambda \end{array}\right) = - \left(\begin{array}{c} \bar{\mathcal{L}}'_{z} \\ \bar{\mathcal{L}}'_{u} \\ \hline \bar{\mathcal{L}}'_{q} \\ \bar{\mathcal{L}}'_{s} \\ \bar{\mathcal{L}}'_{\lambda} \end{array}\right). \tag{5.17}$$

In the IMS framework, the first solution step consists of fixing s^j and λ^{j+1} and then solving the remaining interval BVP (5.2). In contrast, the system (5.17) suggests to fix s^j, λ^{j+1}, as well as the controls q^j. In a first solution step only the state and adjoint variables $u^j = u^j(s^j, q^j)$ and $z^j(s^j, q^j, \lambda^{j+1})$ are computed, which become dependent variables. Note that u^j does not depend on z^j and therefore is independent of λ^{j+1}. This proceeding results in the following intervalwise IVP:

$$(\!(\partial_t u^j, \delta z)\!) + a(u^j)(\delta z) + b(q^j)(\delta z) - (\!(f|_{I_j}, \delta z)\!) + (u^j(\tau_j) - s^j, \delta z(\tau_j)) = 0, \tag{5.18a}$$

$$J_u^{j\prime}(q^j, u^j)(\delta u) - (\!(\partial_t z^j, \delta u)\!) + a'_u(u^j)(\delta u, z^j) + (z^j(\tau_{j+1}) - \lambda^{j+1}, \delta u(\tau_{j+1})) = 0. \tag{5.18b}$$

Equivalently to the IMS case, these equations appear to generate a BVP structure with separated boundary values, as the state equation (5.18a) runs forward and the adjoint equation (5.18b) runs backward in time. In the IMS system (5.2), the three equations are strongly coupled, i. e., after solving the state and adjoint equations, the control equation provides a feedback for the state, thus starting an iterative solution process. In contrast, system (5.18) can be considered as two successive IVP. The equations are only weakly coupled and constitute no optimal control problems, due to the missing intervalwise control equations. Therefore, the state solutions $u^j(s^j, q^j)$ can be computed first and then used to compute the adjoint solutions $z^j(u^j(s^j, q^j); \lambda^{j+1})$, where the latter has to be carried out backward in time. In the DMS context, this IVP formulation ia a suitable starting point. The two local IVP (5.18) correspond to the first two equation blocks of (3.48), $\bar{\mathcal{L}}'_{z^j} = 0$ and $\bar{\mathcal{L}}'_{u^j} = 0$.
After performing the first step, we still have to solve the matching conditions $\bar{\mathcal{L}}'_{s^j} =$

0, $\bar{\mathcal{L}}'_{\lambda^j} = 0$ and the control equations $\bar{\mathcal{L}}'_{q^j} = 0$, which together constitute the second solution step. The resulting system that has to be solved by Newton's method reads

$$\begin{aligned}
(s^0 - u_0, \delta\lambda) &= 0, && \text{(5.19a)}\\
J_q^{0\prime}(q^0, u^0)(\delta q) + b'_q(q^0)(\delta q, z^0(\tau_j; s^0, q^0, \lambda^1)) &= 0, && \text{(5.19b)}\\
(\lambda^j - z^j(\tau_j; s^j, q^j, \lambda^{j+1}), \delta s) &= 0, && \text{(5.19c)}\\
(s^j - u^{j-1}(\tau_j; s^{j-1}, q^{j-1}), \delta\lambda) &= 0, && \text{(5.19d)}\\
J_q^{j\prime}(q^j, u^j)(\delta q) + b'_q(q^j)(\delta q, z^j(\tau_j; s^j, q^j, \lambda^{j+1})) &= 0, && \text{(5.19e)}\\
(\lambda^M, \delta s) &= 0. && \text{(5.19f)}
\end{aligned}$$

As in system (5.3), equations (5.19c) – (5.19e) should be treated as a block for each $j = 1, \ldots, M-1$. Again, the size of the original Newton system (3.50) is reduced, although in the current situation it remains larger than in the IMS framework. This stems from the presence of the intervalwise controls that are distributed both in space and time (see also Remark 5.3 below). Abbreviating (5.19) by $F(\mathbf{s}, \mathbf{q}, \boldsymbol{\lambda}) = 0$ results in Newton's equation

$$\nabla_{(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})} F(\mathbf{s}, \mathbf{q}, \boldsymbol{\lambda}) \cdot \begin{pmatrix} \delta\mathbf{s} \\ \delta\mathbf{q} \\ \delta\boldsymbol{\lambda} \end{pmatrix} = -F(\mathbf{s}, \mathbf{q}, \boldsymbol{\lambda}). \tag{5.20}$$

Altogether, the structure of DMS is again a two-step fixed-point iteration, where the controls and initial values are fixed in the first step for computing (u^j, z^j) before updating $(s^j, q^j, \lambda^{j+1})$ in the second step. The solution process is resumed in the following Algorithm 5.4.

Algorithm 5.4 Direct multiple shooting for PDE governed OCP

Require: Decomposition $I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1}(\tau_j, \tau_{j+1}]$, initial values and controls $\{(s_0^j, q_0^j, \lambda_0^{j+1})_{j=0}^{M-1}\}$.

1: Set $k = 1$.
2: **while** Shooting conditions (5.19) not fulfilled **do**
3: **for** $j = 0$ to $M-1$ **do**
4: Solve intervalwise initial value problems (5.18).
5: **end for**
6: Solve (5.20), compute the update $\{(s_k^j, q_k^j, \lambda_k^{j+1})_{j=0}^{M-1}\}$ of initial values and controls, set $k \leftarrow k+1$.
7: **end while**

Remark 5.3. Before discussing the algorithmic details of DMS, it is important to emphasize that neither of the two approaches is preferable so far. Although the structure of the intervalwise IVP (5.18) is less complex than that of the intervalwise OCP (5.2) and presumably takes less computational effort, the Newton system (5.20) of DMS is larger than the corresponding IMS counterpart (5.4). A comparative example is presented in Section 5.4.

5.2.2 Algorithmic description

In contrast to the IMS case, where the details of the solution of both the intervalwise BVP (5.2) and the system (5.4) of shooting conditions have been presented, the focus is here on step 6 of Algorithm 5.4, i. e., the realization of Newton's method. The solution of the IVP in step 4 is straightforward. It includes solving the state equation on each subinterval because u^j is needed for solving the adjoint equation, which is subsequently solved backward in time. Given this restriction, we turn our attention to Newton's system (5.20). The solution of is similar to the corresponding system in IMS. For better comparability, the following presentation is elaborated in detail.

The shooting system. In Subsection 5.1.2, it is stated that the application of a matrix-free Krylov-Newton method is desirable due to the size of problem (5.4). As the system (5.20) is usually larger than (5.4) due to the presence of the control variables, assembling the matrix is even less advisable in the DMS context. Below, a Newton-GMRES method similar to the one presented in the IMS context is discussed. The main steps are summarized in Algorithm 5.5.
As it is essential for the structure of the implementation, the details of Newton's system (5.20) are presented first. It is more complex than its IMS counterpart, and its explicit form is developed in several steps. Abstractly, it reads

$$\begin{bmatrix} A_0 & B_0 & & \cdots & 0 \\ C_0 & A_1 & B_1 & & \vdots \\ & C_1 & \ddots & & \\ \vdots & & & A_{M-2} & B_{M-2} \\ 0 & \cdots & & C_{M-2} & A_{M-1} \end{bmatrix} \begin{bmatrix} \delta y^0 \\ \delta y^1 \\ \vdots \\ \delta y^{M-2} \\ \delta y^{M-1} \end{bmatrix} = - \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_{M-2} \\ F_{M-1} \end{bmatrix}, \tag{5.21}$$

which appears to be similar to (5.10). For $j = 0, \ldots, M-1$, the solution subvectors are given as $\delta y^j := (\delta s^j, \delta q^j, \delta \lambda^{j+1})^\top$, and on the function space level, the submatrices are given as

$$A_j = \begin{bmatrix} I & 0 & 0 \\ b_q'(q^j)(\cdot, z_s^{j\prime}) & J_{qq}''(q^j, u^j)(\cdot) + b_q'(q^j)(\cdot, z_q^{j\prime}) + b_{qq}''(q^j)(\cdot, z^j) & b_q'(q^j)(\cdot, z_\lambda^{j\prime}) \\ 0 & 0 & I \end{bmatrix},$$

$$B_j = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -z_s^{j+1\prime}(\tau_{j+1}) & -z_q^{j+1\prime}(\tau_{j+1}) & -z_\lambda^{j+1\prime}(\tau_{j+1}) \end{bmatrix}, \quad C_j = \begin{bmatrix} -u_s^{j\prime}(\tau_{j+1}) & -u_q^{j\prime}(\tau_{j+1}) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The righthand side subvectors are explicitly written in the form

$$
\begin{aligned}
F_0 &= \begin{bmatrix} s^0 - u_0 \\ J_q'(q^0, u^0)(\cdot) + b_q'(q^0)(\cdot, z^0(s^0, q^0, \lambda^1)) \\ \lambda^1 - z^1(\tau_1; s^1, \lambda^2) \end{bmatrix}, \\
F_j &= \begin{bmatrix} s^j - u^{j-1}(\tau_j; s^{j-1}, q^{j-1}) \\ J_q'(q^j, u^j)(\cdot) + b_q'(q^j)(\cdot, z^j(s^j, q^j, \lambda^{j+1})) \\ \lambda^{j+1} - z^{j+1}(\tau_{j+1}; s^{j+1}, q^{j+1}, \lambda^{j+2}) \end{bmatrix}, \quad (j = 1, \ldots, M-2), \\
F_{M-1} &= \begin{bmatrix} s^{M-1} - u^{M-2}(\tau_{M-1}; s^{M-2}, \lambda^{M-1}) \\ J_q'(q^{M-1}, u^{M-1})(\cdot) + b_q'(q^{M-1})(\cdot, z^{M-1}(s^{M-1}, q^{M-1}, \lambda^M)) \\ \lambda^M \end{bmatrix}.
\end{aligned}
$$

The Jacobian $\nabla_{(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})} F(\mathbf{s}, \mathbf{q}, \boldsymbol{\lambda})$ of (5.19) involves derivatives $u_s^{j\prime}, u_q^{j\prime}$ of u^j w. r. t. s^j and q^j as well as derivatives $z_s^{j\prime}, z_\lambda^{j\prime}, z_q^{j\prime}$ of z^j w. r. t. s^j, λ^{j+1} and q^j. These derivatives, the so-called sensitivities, are obtained by solving five additional (linearized) IVP, the sensitivity or variational equations, for $j \in \{0, \ldots, M-1\}$. First, (5.18a) where $u^j = u^j(s^j, q^j)$ is differentiated with respect to s^j in direction δs and with respect to q^j in direction δq to obtain the equations

$$
\begin{aligned}
(\!(\partial_t u_s^{j\prime}, \varphi)\!) + a_u'(u^j)(u_s^{j\prime}, \varphi) + (u_s^{j\prime}(\tau_j) - \delta s^j, \varphi(\tau_j)) &= 0, && \text{(5.22a)} \\
(\!(\partial_t u_q^{j\prime}, \varphi)\!) + a_u'(u^j)(u_q^{j\prime}, \varphi) + b_q'(q^j)(\delta q^j, \varphi) + (u_q^{j\prime}(\tau_j), \varphi(\tau_j)) &= 0, && \text{(5.22b)}
\end{aligned}
$$

which have to hold for all $\varphi \in X^j$. Solving these problems for given initial data $\delta s^{j,0}$ and $\delta q^{j,0}$ results in $u_s^{j\prime}, u_q^{j\prime}$. These sensitivities can be inserted into the following three IVP obtained by differentiating the adjoint equation (5.18b) with respect to all its arguments in corresponding directions. The following equations must hold for all $\psi \in X^j$:

$$
\begin{aligned}
J_{uu}^{j\prime\prime}(q^j, u^j)(u_s^{j\prime}, \psi) - (\!(\partial_t z_s^{j\prime}, \psi)\!) + a_{uu}''(u^j)(u_s^{j\prime}, \psi, z^j) & \\
+ a_u'(u^j)(\psi, z_s^{j\prime}) + (z_s^{j\prime}(\tau_{j+1}), \psi(\tau_{j+1})) &= 0, && \text{(5.23a)} \\
J_{uu}^{j\prime\prime}(q^j, u^j)(u_q^{j\prime}, \psi) - (\!(\partial_t z_q^{j\prime}, \psi)\!) + a_{uu}''(u^j)(u_q^{j\prime}, \psi, z^j) & \\
+ a_u'(u^j)(\psi, z_q^{j\prime}) + (z_q^{j\prime}(\tau_{j+1}), \psi(\tau_{j+1})) &= 0, && \text{(5.23b)} \\
-(\!(\partial_t z_\lambda^{j\prime}, \psi)\!) + a_u'(u^j)(\psi, z_\lambda^{j\prime}) + (z_\lambda^{j\prime}(\tau_{j+1}) - \delta\lambda^{j+1}, \psi(\tau_{j+1})) &= 0. && \text{(5.23c)}
\end{aligned}
$$

Solving these problems with initial data $\delta\lambda^{j+1,0}$ results in a complete set of sensitivities, but only with respect to the chosen initial values $(\delta s^{j,0}, \delta q^{j,0}, \delta\lambda^{j+1,0})$. In order to assemble $\nabla_{(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})} F$ explicitly, the sensitivity equations have to be solved for a complete basis of $\bigcup_{j=0}^{M-1} [H \times Q^j \times H]$, which is numerically expensive for fine temporal or spatial discretizations. Therefore, we choose a matrix-free approach where we handle Newton's system (5.20) with an iterative solver, for which we choose in our case, due to the asymmetric structure of the matrix, a GMRES method. We then have to solve the sensitivity equations only once per GMRES iteration. The adjoint approach thus avoids assembling the Jacobian and operates on the matrix-vector product $\nabla_{(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})} F(\mathbf{s}, \mathbf{q}, \boldsymbol{\lambda}) \cdot (\delta\mathbf{s}, \delta\mathbf{q}, \delta\boldsymbol{\lambda})^\top$ instead.

This matrix-vector product, the left-hand side of (5.20), has the concrete form

$$\nabla_{(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})}F(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})\cdot\begin{bmatrix}\delta\mathbf{s}\\ \delta\mathbf{q}\\ \delta\boldsymbol{\lambda}\end{bmatrix}=\begin{bmatrix}\delta s^0\\ J_{qq}^{0\prime\prime}(q^0,u^0)(\delta q^0)+b_{qq}''(q^0)(\delta q^0,z^0)\\ \quad +b_q'(q^0)[z_s^{0\prime}(\delta s^0)+z_q^{0\prime}(\delta q^0)+z_\lambda^{0\prime}(\delta\lambda^0)]\\ \hline \delta\lambda^j-z_s^{j\prime}(\tau_j;\delta s^j)-z_q^{j\prime}(\tau_j;\delta q^j)-z_\lambda^{j\prime}(\tau_j;\delta\lambda^{j+1})\\ \delta s^j-u_s^{j-1\prime}(\tau_j;\delta s^{j-1})-u_\lambda^{j-1\prime}(\tau_j;\delta\lambda^j)\\ J_{qq}^{j\prime\prime}(q^j,u^j)(\delta q^j)+b_{qq}''(q^j)(\delta q^j,z^j)\\ \quad +b_q'(q^j)[z_s^{j\prime}(\delta s^j)+z_q^{j\prime}(\delta q^j)+z_\lambda^{j\prime}(\delta\lambda^j)]\\ \hline \delta\lambda^M\end{bmatrix},\qquad(5.24)$$

where the middle part has to be interpreted for $j=1,\dots,M-1$ (cf. equations (5.19)). This enables the formulation of the following Algorithm 5.5 which yields the details of step 6 of the above DMS algorithm:

Algorithm 5.5 Solution of the DMS shooting system (matrix-free approach)

Require: Shooting variables $(\mathbf{s}_k,\boldsymbol{\lambda}_k)$ and controls $\mathbf{q}_k$, intervalwise OCP solutions u^j,z^j

1: Build up residual $-F(\mathbf{s}_k,\mathbf{q}_k,\boldsymbol{\lambda}_k)$.
2: Set $i=0$, prescribe tolerance TOL and choose $(\delta\mathbf{s}_k^{(0)},\delta\mathbf{q}_k^{(0)},\delta\boldsymbol{\lambda}_k^{(0)})$.
3: **while** $\|\nabla F(\mathbf{s}_k,\mathbf{q}_k,\boldsymbol{\lambda}_k)(\delta\mathbf{s}_k^{(i)},\delta\mathbf{q}_k^{(i)},\delta\boldsymbol{\lambda}_k^{(i)})+F(\mathbf{s}_k,\mathbf{q}_k,\boldsymbol{\lambda}_k)\|>TOL$ **do**
4: Compute matrix-vector product $\nabla F(\mathbf{s}_k,\mathbf{q}_k,\boldsymbol{\lambda}_k)(\delta\mathbf{s}_k^{(i)},\delta\mathbf{q}_k^{(i)},\delta\boldsymbol{\lambda}_k^{(i)})$ by solving the state and adjoint sensitivity equations (5.22) and (5.23).
5: Solve system $\nabla F(\mathbf{s}_k,\mathbf{q}_k,\boldsymbol{\lambda}_k)(\delta\mathbf{s}_k^{(i)},\delta\mathbf{q}_k^{(i)},\delta\boldsymbol{\lambda}_k^{(i)})=-F(\mathbf{s}_k,\mathbf{q}_k,\boldsymbol{\lambda}_k)$ by a Newton-GMRES method. This requires the renewed solution of (5.22) and (5.23) in each iteration.
6: **end while**
7: Set $k\leftarrow k+1$ and $\mathbf{s}_{k+1}=\mathbf{s}_k+\delta\mathbf{s}_k^{end},\mathbf{q}_{k+1}=\mathbf{q}_k+\delta\mathbf{q}_k^{end},\boldsymbol{\lambda}_{k+1}=\boldsymbol{\lambda}_k+\delta\boldsymbol{\lambda}_k^{end}$.

Remark 5.4. In the presentation of the substructures of the linear system (5.21) above, it is emphasized that all subvectors and -matrices are formulated on the continuous (function space) level. In the IMS case in Subsection 5.1.2, none of the shooting variables s^j and λ^{j+1} is distributed in the time variable. Hence, a distinction between the continuous and the temporally discrete cases is not necessary. In the DMS framework, the controls q^j are part of the shooting system, and they are usually time-dependent and therefore have to be temporally discretized (see the discussion in Subsection 4.1.4). As the description of the matrices A_j,B_j and C_j on the continuous level subsequent to (5.21) can easily lead to a misestimation of their dimensionality, this subsection concludes with a presentation of their temporally discrete counterpart. Therefore, we assume $\bar{I}_j=\{\tau_j\}\cup(\tau_j,\tau_{j+1}]$ to be decomposed as in (4.1). For simplicity, the number N of time steps is assumed the same on all shooting intervals. Differentiation of u^j and z^j w. r. t. s^j,q^j and λ^{j+1} results in the need to solve additional linearized IVP, as seen above in (5.22) and (5.23). The controls,

as well as all sensitivity solutions, are spatially distributed functions in each timepoint t_i^j. For the following presentation, the control part of the functional on I_j is given as usual by $\alpha(\!(q^j, q^j)\!)$. Furthermore, the control operator $b(q^j)(z^j)$ is linear and given as the scalar product $(\!(q^j, z^j)\!)$. This leads to the following concrete block matrices:

$$A_j = \begin{bmatrix} I & 0 & 0 & 0 & \cdots & 0 & 0 \\ z_{s^j}^{j\prime}(t_0^j) & \alpha I + z_{q^j}^{j\prime}(t_0^j) & 0 & 0 & \cdots & 0 & z_{\lambda^{j+1}}^{j\prime}(t_0^j) \\ z_{s^j}^{j\prime}(t_1^j) & 0 & \alpha I + z_{q^j}^{j\prime}(t_1^j) & 0 & \cdots & 0 & z_{\lambda^{j+1}}^{j\prime}(t_1^j) \\ \vdots & & & \ddots & & & \vdots \\ z_{s^j}^{j\prime}(t_N^j) & 0 & 0 & 0 & \cdots & \alpha I + z_{q^j}^{j\prime}(t_N^j) & z_{\lambda^{j+1}}^{j\prime}(t_N^j) \\ 0 & 0 & 0 & 0 & \cdots & 0 & I \end{bmatrix}$$

constitute the diagonal blocks, whereas the blocks on the two secondary diagonals are given by

$$B_j = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ -z_{s^{j+1}}^{j+1\prime}(\tau_{j+1}) & -z_{q^{j+1}}^{j+1\prime}(\tau_{j+1}) & 0 & \cdots & 0 & -z_{\lambda^{j+2}}^{j+1\prime}(\tau_{j+1}) \end{bmatrix},$$

$$C_j = \begin{bmatrix} -u_{s^j}^{j\prime}(\tau_{j+1}) & 0 & 0 & \cdots & -u_{q^j}^{j\prime}(\tau_{j+1}) & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Each entry of one such block matrix stands for an $R \times R$ block matrix itself, where $R = \dim V_h^s$. Analogously, one block vector of the righthand side is given by

$$F_j = \begin{bmatrix} s^j - u^{j-1}(\tau_j; s^{j-1}, q^{j-1}) \\ \alpha q^j(t_0^j) + z^j(t_0^j; s^j, q^j, \lambda^{j+1}) \\ \alpha q^j(t_1^j) + z^j(t_1^j; s^j, q^j, \lambda^{j+1}) \\ \vdots \\ \alpha q^j(t_N^j) + z^j(t_N^j; s^j, q^j, \lambda^{j+1}) \\ \lambda^{j+1} - z^{j+1}(\tau_{j+1}; s^{j+1}, q^{j+1}, \lambda^{j+2}) \end{bmatrix}.$$

Note that for the matrices, the relation $t_N^j = \tau_{j+1} = t_0^{j+1}$ is implicitly used, and a function q^j on the temporally discrete level stands for a vector $(q^j(t_0^j), q^j(t_1^j), \cdots, q^j(t_N^j))$. Thus, the whole Jacobian $\nabla_{(\mathbf{s},\mathbf{q},\boldsymbol{\lambda})} F(\mathbf{s}, \mathbf{q}, \boldsymbol{\lambda})$ in the DMS case has dimension $[(N+3)MR] \times [(N+3)MR]$, where N is the number of time steps per shooting interval. Recall that the dimension of the corresponding Jacobian $\nabla_{(\mathbf{s},\boldsymbol{\lambda})} F(\mathbf{s}, \boldsymbol{\lambda})$ in the IMS framework is $2MR \times 2MR$.

Preconditioned DMS. As provided in Subsection 5.1.2, for an increasing number of shooting intervals the conditioning of the Newton system for solving the continuity conditions deteriorates. In particular, the condition number of the respective Jacobian ∇F grows (cf. the discussion of Example 2.1 in Section 2.2). The discussion of a symmetric Gauss-Seidel type preconditioner in the IMS framework followed Hesse [52]. The idea is originally based on former considerations of Comas [26] and Heinkenschloss [50]. The latter authors already design different preconditioning approaches for DMS methods (or, more generally, time domain decomposition methods). Their DMS symmetric Gauss-Seidel preconditioner is similar to the IMS case in Subsection 5.1.2, but is more complicated because the diagonal blocks A_j are now different from the identity and have to be inverted. We omit the details and instead content ourselves with referring to the literature. Observing the current state of research in time domain decomposition preconditioning, there always seems to be a trade-off between efficient preconditioning and parallelizability. Efficient preconditioners often corrupt the potential of time domain decomposition methods for parallel computing, whereas preconditioners that respect parallelization are usually far less efficient. Therefore, this research field still has promising perspectives.

5.3 Two variants of DMS

The separating line between direct and indirect methods for OCP is not clearly defined. In the optimal control community, the following may be regarded as a common denominator: In direct methods, the problem is first discretized (the result of which is often called a nonlinear programming problem (NLP)), and the optimization follows on the discrete level (*'first discretize then optimize'*). In contrast, indirect methods are based on the derivation of the equations forming the first order optimality conditions which are discretized only afterwards (*'first optimize then discretize'*). For a more detailed presentation of these two techniques, see, e. g., the textbook by Hinze et al. [59], as well as the introductions of both Albersmeyer [1] and Rao [96]. The distinction reflects a paradigm known from the calculus of variations. There, indirect methods are based on Pontryagin's maximum principle. A detailed presentation of direct approaches is found in the textbook of Dacorogna [29].
Classical multiple shooting methods for OCP as presented in Section 2.3 are classified as direct or indirect according to the above general criterion. In this regard, the DMS method derived in Section 5.2 seems to be rather an indirect method, as it is based on the KKT system (3.48), i. e., the optimality conditions have been stated before shooting comes into play. In addition, the whole discussion of Section 5.2 is done on the continuous function space level, whereas the above classification requires a discretization in case of a direct method. A third indicator is the presence of an adjoint equation in the presented DMS algorithm, which has for a long time been regarded as artificial from the viewpoint of direct methods (see, however, the recent work by Albersmeyer [1] and Beigel [8]).
This section explains why, although the method from Section 5.2 does not fit into the usual framework of direct methods, it is nevertheless denoted as a direct multiple shooting method. In Subsection 5.3.1, the 'classical' DMS approach is tailored to the parabolic situation. Then Subsection 5.3.2 illustrates that this approach, relying on a reduced formulation of

the extended optimal control problem (3.43) – (3.44), is equivalent to the variant of DMS presented in the previous section. The difference between the two DMS methods is a more complicated example for the well-known dichotomy of *sensitivity approaches* and *adjoint approaches* that are used to generate derivatives of a reduced cost functional. The principle is already known from Section 3.3. For a thorough discussion of this dichotomy, see Hinze et al. [59] for the global OCP (3.3) – (3.4).

5.3.1 DMS based on a reduced form of the extended OCP

The DMS method which was developed in the 1980s by Bock and his co-workers [11, 13, 15]) is now embedded into the context of parabolic OCP. In Leineweber [73], a detailed summary of the overall solution process for ODE control problems is presented, discussing all involved algorithms and proposing alternatives. More recently developed techniques that allow for handling parabolic PDE in the method of lines (MOL) framework can be found in the work of Potschka [94].
DMS methods for problem (3.43) – (3.44) are usually based on a reformulation of this extended OCP in terms of the primal shooting variables s^j and the intervalwise controls q^j. Therefore, it yields $u^j = u^j(q^j, s^j)$, but in contrast to Section 5.2, where the system of optimality conditions is split into two parts according to (5.17), these dependencies are now induced from the very beginning, i. e., before the optimization process has taken place. Pursuing this strategy, the minimization problem results in

$$\min_{(\mathbf{q},\mathbf{s})} \ \overline{J}(\mathbf{q},\mathbf{s}) := \sum_{j=0}^{M-1} J^j(q^j, u^j(q^j, s^j)) \tag{5.25a}$$

$$\text{s. t.} \qquad s^0 - u_0 = 0, \tag{5.25b}$$

$$s^{j+1} - u^j(\tau_{j+1}; q^j, s^j) = 0, \tag{5.25c}$$

where (5.25c) comprises the continuity conditions for $j = 0, \ldots, M-1$. System (5.25) is called a reduced formulation of the extended OCP (3.43) – (3.44). Reduced approaches are, as provided in Section 3.3, a common technique in OCP. In fact, a reduced approach has been applied in Algorithm 5.2 for solving the subinterval BVP. A detailed description is also given in the textbook by Hinze et al. [59]. In Section 3.3, the variable u in (3.3) – (3.4) is assumed to depend on q by using the implicit function theorem. Here, we deal with an extended set of variables, meaning that the problem is reduced to $(\mathbf{q}, \mathbf{s})$ rather than to $\mathbf{q}$ only. Problem (5.25) is formulated in terms of these independent variables and relies upon the IVP

$$e^j(q^j, s^j, u^j(q^j, s^j)) = \begin{pmatrix} \partial_t u^j(q^j, s^j) + \mathcal{A}(u^j(q^j, s^j)) + \mathcal{B}(q^j) - f|_{I_j} \\ u^j(\tau_j; q^j, s^j) - s^j \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.26}$$

having been solved on all subintervals I_j for $j = 0, \ldots, M-1$. We assume unique solvability of the subinterval problems which implies the existence of a solution operator mapping $Q^j \times H$ to X^j. In (5.26), $e^j(q^j, s^j, u^j(q^j, s^j))$ is an intervalwise counterpart of the abstract side condition (3.2) which is, in contrast to the preceding sections, again strongly formulated.

This abstract notation helps us to keep the proof of the equivalence result stated in the next subsection short.

Remark 5.5. In the reformulation (5.25) – (5.26) of the extended OCP, the local control variable $q^j(x,t)$ is a function of both spatial variables x and time t. As provided in Subsection 2.3.3 on DMS methods for ODE control problems depending only on t, the control is usually interpreted as a piecewise polynomial of order $p \leq 3$ on the subintervals I_j, i. e., $q^j \equiv q^j(q_0^j, \cdots, q_p^j)$. Example 2.2 shows that this parameterization of the control saves a large amount of computing time and storage. There, a small set of control parameters per shooting interval I_j is opposed to a number of control values on each I_j determined by the control discretization, which is usually much finer than the mentioned parameterization. Furthermore, so-called condensing techniques which reduce the shooting system to the control variables are frequently employed. However, they are not efficient if q is discretized on a similarly fine level as the state u. Reducing the control to smaller spaces by parameterization leads to suboptimal solutions of the given control problems, as local features of the control cannot be resolved on coarse grids. Furthermore, in the PDE case where control functions $q(x,t)$ are generally distributed in space and time, we usually cannot determine a meaningful parameterization without losing important structural information on q.

The Lagrange functional for the reduced problem (5.25), introducing a Lagrange multiplier $\boldsymbol{\lambda} = (\lambda^j)_{j=0}^M \in H^{M+1}$, is given by

$$\mathcal{L}(\mathbf{q}, \mathbf{s}, \boldsymbol{\lambda}) = \overline{J}(\mathbf{q}, \mathbf{s}) + (s^0 - u_0, \lambda^0) + \sum_{j=0}^{M-1} (s^{j+1} - u^j(\tau_{j+1}; q^j, s^j), \lambda^{j+1}). \tag{5.27}$$

Differentiating this functional as usual with respect to its arguments $(\mathbf{q}, \mathbf{s}, \boldsymbol{\lambda})$, we obtain the (reduced) optimality system (where $j \in \{0, \ldots, M-1\}$ in equations (5.28b), (5.28c) and (5.28e)):

$$\begin{aligned}
\mathcal{L}'_{\lambda^0}(\delta\lambda) &= (s^0 - u_0, \delta\lambda), && (5.28a)\\
\mathcal{L}'_{\lambda^j}(\delta\lambda) &= (s^{j+1} - u^j(\tau_{j+1}; q^j, s^j), \delta\lambda), && (5.28b)\\
\mathcal{L}'_{s^j}(\delta s) &= \langle J_u^{j\prime}, u_s^{j\prime}(\delta s)\rangle_{X^{j*}\times X^j} + (\lambda^j, \delta s) - (\lambda^{j+1}, u_s^{j\prime}[\tau_{j+1}](\delta s)), && (5.28c)\\
\mathcal{L}'_{s^M}(\delta s) &= (\lambda^M, \delta s), && (5.28d)\\
\mathcal{L}'_{q^j}(\delta q) &= \langle J_q^{j\prime}, \delta q\rangle_{Q^{j*}\times Q^j} + \langle J_u^{j\prime}, u_q^{j\prime}(\delta q)\rangle_{X^{j*}\times X^j} - (\lambda^{j+1}, u_q^{j\prime}[\tau_{j+1}](\delta q)). && (5.28e)
\end{aligned}$$

Here, the notation $u_{q/s}^{j\prime}[\tau_{j+1}]$ states that the respective solution obtained by application of the operator $u_{q/s}^{j\prime}$ is evaluated at time-point τ_{j+1}. The classical DMS method consists in the solution of system (5.28). In the framework of ODE optimal control, Leineweber [73] gives a detailed description of SQP methods that solve (5.28) without employing an adjoint equation. Therefore, either the Newton matrix has to be assembled, or additional sophisticated algorithms are needed to circumvent this matrix assembly. An alternative matrix-free SQP approach has been proposed by Ulbrich [110]. This procedure corresponds to the sensitivity approach for generating derivative information that is needed during the

solution process. In PDE optimal control, the sensitivity approach which was introduced in Section 3.3 is usually too expensive, because one has to solve an additional linearized problem for each basis vector δq of the (discrete) control space (see Hinze et al. [59] or Meidner [84]).

5.3.2 Equivalence of the two DMS approaches

If we compare the two DMS variants presented in Section 5.2 and Subsection 5.3.1, a central distinction that strongly influences the structure of the solution process becomes obvious. The DMS variant discussed in Section 5.2 is, equal to the IMS approach of Section 5.1, based on the full optimality system (3.48), including the adjoint part. In the classical DMS method from Subsection 5.3.1, no adjoint problem occurs. Therefore, it is a priori not clear whether Subsection 5.2 describes a DMS method in the 'classical' sense of Subsection 5.3.1.

Remark 5.6. Generally, modern implementations of DMS for ODE optimal control problems, which are capable of handling parabolic OCP by transforming the PDE side condition into a huge ODE system via the MOL approach, make use of adjoint methods for sensitivity generation. These constitute a suitable alternative to the above described adjoint-free approach (see, e. g., Albersmeyer [1] or Beigel [8]). They often compute the adjoint equations by automatic differentiation (which may be difficult to derive by hand in case of large and highly nonlinear ODE systems).

The following theorem states the main result of this section, showing the equivalence of the two DMS approaches. Moreover, the proof reveals that the seemingly different DMS variant of Subsection 5.2 is a reformulation of classical DMS by means of an adjoint approach for sensitivity generation. It is performed in an abstract function space setting, ensuring that the argumentation is not affected by discretization.

Theorem 5.1. *The solution of the reduced formulation* (5.25) *of the modified OCP* (3.43) – (3.44) *by an adjoint approach leads to the non-classical DMS method introduced in Subsection 5.2.*

The following outline prepares the proof of Theorem 5.1 (as the interrelations in terms of the equations might be confusing, we illustrate them again in Figure 5.1). Classical DMS for problem (5.25) relies upon the solution of (5.26) and necessitates solving (5.28). Analogously, the DMS approach from Subsection 5.2 for problem (3.43) – (3.44) relies upon the solution of (5.18) and necessitates solving (5.19). Comparing the two settings, the following correspondencies are evident: (5.26) is the strong formulation of (5.18a), and (5.28a), (5.28b) and (5.28d) are identical to (5.19a), (5.19d) and (5.19f), respectively. It is thus our goal to derive the adjoint equation (5.18b), the belonging continuity conditions (5.19c) and the control equations (5.19b) and (5.19e) from (5.28c) and (5.28e). To achieve this, the ideas and techniques of Section 1.6 from the book of Hinze et al. [59] are extended to the more complex multiple shooting situation.

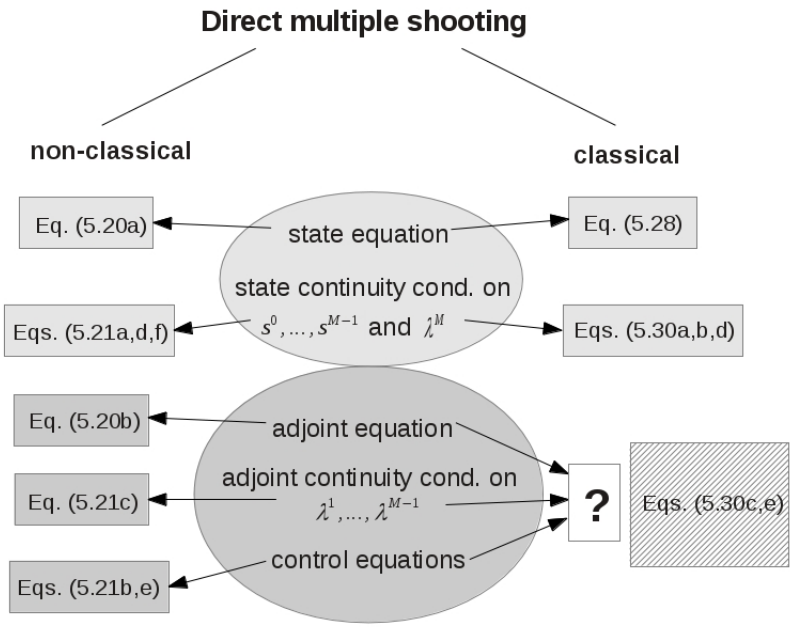

Figure 5.1. Illustration of the relationship between non-classical and classical direct multiple shooting.

Proof. In the following discussion, we make use of adjoint operators $u_q^{j\prime *} : X^{j*} \to Q^{j*}$ and $u_s^{j\prime *} : X^{j*} \to H^* \equiv H$ as well as their evaluations at the subinterval endpoint, $u_q^{j\prime *}[\tau_{j+1}] : H \equiv H^* \to Q^{j*}$ and $u_s^{j\prime *}[\tau_{j+1}] : H \equiv H^* \to H^* \equiv H$. The adjoint operators correspond to the differential operators $u_q^{j\prime} : Q^j \to X^j$ and $u_s^{j\prime} : H \to X^j$. By means of these adjoint operators, equations (5.28c) and (5.28e) can be rewritten in an abstract adjoint form:

$$\mathcal{L}'_{s^j}(\delta s) = (u_s^{j\prime *}(J_u^{j\prime}), \delta s) + (\lambda^j, \delta s) - (u_s^{j\prime *}[\tau_{j+1}](\lambda^{j+1}), \delta s), \tag{5.27c*}$$

$$\begin{aligned}\mathcal{L}'_{q^j}(\delta q) = {} & \langle J_q^{j\prime}, \delta q\rangle_{Q^{j*}\times Q^j} + \langle u_q^{j\prime *}(J_u^{j\prime}), \delta q\rangle_{Q^{j*}\times Q^j} \\ & - \langle u_q^{j\prime *}[\tau_{j+1}](\lambda^{j+1}), \delta q\rangle_{Q^{j*}\times Q^j}.\end{aligned} \tag{5.27e*}$$

Now the adjoint operators are discussed on an abstract level, which enables a clear presentation of the formal framework. Differentiation of the interval state equations (5.26) w. r. t. q^j in direction δq and w. r. t. s^j in direction δs yields (for brevity, the arguments $(q^j, s^j, u^j(q^j, s^j))$ are omitted)

$$e_u^{j\prime}(\delta u_q) = -e_q^{j\prime}(\delta q), \quad e_u^{j\prime}(\delta u_s) = -e_s^{j\prime}(\delta s). \tag{5.29}$$

As usual, the abbreviations δu_q and δu_s for the sensitivities $u_q^{j\prime}(\delta q)$ and $u_s^{j\prime}(\delta s)$ are used, respectively. It is a standard assumption that $e_u^{j\prime}$ has a bounded inverse, which permits application of the implicit function theorem to obtain

$$u_q^{j\prime} = -(e_u^{j\prime})^{-1} \circ e_q^{j\prime}, \quad u_s^{j\prime} = -(e_u^{j\prime})^{-1} \circ e_s^{j\prime}.$$

The definition and calculation rules of adjoint operators give us the following abstract representation, where the inverse of the adjoint is denoted by the superscript $-*$:

$$u_q^{j\prime *} = -e_q^{j\prime *} \circ (e_u^{j\prime})^{-*}, \quad u_s^{j\prime *} = -e_s^{j\prime *} \circ (e_u^{j\prime})^{-*}.$$

Inserting these expressions for $u_s^{j\prime *}$ and $u_q^{j\prime *}$ into the corresponding terms of (5.27c*) and (5.27e*) results in

$$\begin{aligned}
(u_s^{j\prime *}(J_u^{j\prime}), \delta s) &= -(e_s^{j\prime *}((e_u^{j\prime})^{-*}(J_u^{j\prime})), \delta s), && \text{(5.30a)}\\
(u_s^{j\prime *}[\tau_{j+1}](\lambda^{j+1}), \delta s) &= -(e_s^{j\prime *}((e_u^{j\prime})^{-*}[\tau_{j+1}](\lambda^{j+1})), \delta s), && \text{(5.30b)}\\
\langle u_q^{j\prime *}(J_u^{j\prime}), \delta q\rangle_{Q^{j*}\times Q^j} &= -\langle e_q^{j\prime *}((e_u^{j\prime})^{-*}(J_u^{j\prime})), \delta q\rangle_{Q^{j*}\times Q^j}, && \text{(5.30c)}\\
\langle u_q^{j\prime *}[\tau_{j+1}](\lambda^{j+1}), \delta q\rangle_{Q^{j*}\times Q^j} &= -\langle e_q^{j\prime *}((e_u^{j\prime})^{-*}[\tau_{j+1}](\lambda^{j+1})), \delta q\rangle_{Q^{j*}\times Q^j}. && \text{(5.30d)}
\end{aligned}$$

It is now important that we apply both operators $e_s^{j\prime *}$ and $e_q^{j\prime *}$ to the same argument $(e_u^{j\prime})^{-*}(J_u^{j\prime})$ in (5.30a) and (5.30c). This also holds true for $(e_u^{j\prime})^{-*}[\tau_{j+1}](\lambda^{j+1})$ in (5.30b) and (5.30d). We are now in a position to define the variables $z_J^j := -(e_u^{j\prime})^{-*}(J_u^{j\prime})$ and $z_\lambda^j := (e_u^{j\prime})^{-*}[\tau_{j+1}](\lambda^{j+1})$, which fulfil the following equations, respectively:

$$e_u^{j\prime *}(z_J^j) = -J_u^{j\prime}, \qquad e_u^{j\prime *}[\tau_{j+1}](z_\lambda^j) = \lambda^{j+1}. \tag{5.31}$$

These are the formal adjoint equations; below it becomes clear that, due to the linearity of the operator $e_u^{j\prime *}$ and a superposition principle, they can be combined into one equation. Therefore, z_J^j is interpreted as a solution to a problem with homogeneous initial value and non-homogeneous right-hand side $-J_u^{j\prime}$, and z_λ^j as a solution to a problem with homogeneous right-hand side and non-homogeneous initial value λ^{j+1}.
The starting point for further investigations is the weak formulation of (5.26), given as

$$((\partial_t u^j, \varphi)) + a(u^j)(\varphi) + b(q^j)(\varphi) - ((f|_{I_j}, \varphi)) + (u^j(\tau_j) - s^j, \varphi(\tau_j)) = 0. \tag{5.32}$$

The differential operator $u_q^{j\prime} : Q^j \to X^j$ mentioned above is at the same time the solution operator of the following linearized equation (the so-called sensitivity or tangent equation which is the derivative of (5.32) with respect to q^j in direction δq):

$$((\partial_t \delta u_q, \varphi)) + a'_u(u^j)(\delta u_q, \varphi) + (\delta u_q(\tau_j), \varphi(\tau_j)) = -b'_q(q^j)(\delta q, \varphi). \tag{5.33}$$

In complete analogy, $u_s^{j\prime} : H \to X^j$ is the solution operator of a second sensitivity equation, given as the derivative of (5.32) w. r. t. s^j in direction δs:

$$((\partial_t \delta u_s, \varphi)) + a'_u(u^j)(\delta u_s, \varphi) + (\delta u_s(\tau_j), \varphi(\tau_j)) = (\delta s, \varphi(\tau_j)). \tag{5.34}$$

Here, the respective solution variables of both sensitivity equations are denoted by δu_q and δu_s. They correspond to the formal equations (5.29).
Now the formal adjoint equations (5.31) can be concretized. As provided above, the application of the adjoint operators $u_q^{j\prime *}$ or $u_s^{j\prime *}$ to a functional $J_u^{j\prime} \in X^{j\,*}$ (or of the timepoint evaluation operators $u_q^{j\prime *}[\tau_{j+1}]$ or $u_s^{j\prime *}[\tau_{j+1}]$ to the shooting variables λ^{j+1}) corresponds to carrying out the following two steps:

1. Solve the adjoint equation $e_u^{j\prime *}(z_J^j) = -J_u^{j\prime}$ (or $e_u^{j\prime *}[\tau_{j+1}](z_\lambda^j) = \lambda^{j+1}$).
2. Apply the adjoint operators $e_q^{j\prime *}$ and $e_s^{j\prime *}$ to the solution z_J^j (or z_λ^j).

We obtain that the first step leads to the adjoint equation (5.18b), whereas the second step yields the continuity conditions for the adjoint equation and the control equations (remember that it is the objective of the proceeding to derive these equations from (5.28c) and (5.28e)). The general form of the adjoint equation, corresponding to both (5.33) and (5.34), is given by

$$-((\partial_t \delta u_{q/s}^*, \psi)) + a_u'(u^j)(\delta u_{q/s}^*, \psi) + (\delta u_{q/s}^*(\tau_{j+1}), \psi(\tau_{j+1})) = \mathrm{rhs}(\psi). \tag{5.35}$$

Here, the term $\mathrm{rhs}(\psi)$ which determines the dynamics of the linearized equation represents either a distributed source term or an initial condition prescribed at the subinterval endpoint. The abstract situation comprises the following two adjoint equations (where the abstract adjoint variables δu_q^* and δu_s^* have been suitably replaced by z_J^j and z_λ^j):

$$-((\partial_t z_J^j, \psi)) + a_u'(u^j)(z_J^j, \psi) + (z_J^j(\tau_{j+1}), \psi(\tau_{j+1})) = -J_u^{j\prime}(q^j, u^j)(\psi), \tag{5.36a}$$
$$-((\partial_t z_\lambda^j, \psi)) + a_u'(u^j)(z_\lambda^j, \psi) + (z_\lambda^j(\tau_{j+1}), \psi(\tau_{j+1})) = (\lambda^{j+1}, \psi(\tau_{j+1})). \tag{5.36b}$$

Obviously, both components of (5.36) are fully linear, because only the derivatives of the nonlinear operators $a(\cdot)(\cdot)$ and $J^j(\cdot)$ enter the equations. Thus equations (5.36a) and (5.36b) can be merged into one single equation by defining $z^j := z_J^j - z_\lambda^j$, i. e., by subtraction of the equations. The resulting final adjoint equation reads

$$\begin{aligned} J_u^{j\prime}(q^j, u^j)(\psi) - ((\partial_t z^j, \psi)) + a_u'(u^j)(z^j, \psi) & \\ +(z^j(\tau_{j+1}) - \lambda^{j+1}, \psi(\tau_{j+1})) &= 0. \end{aligned} \tag{5.37}$$

A comparison of (5.18b) and (5.37) shows that, by substituting δu by the test function ψ, our first objective is achieved. It consists in the introduction of the adjoint equation into the reduced DMS method by means of an adjoint approach to sensitivity generation.
Finally, the second step of the above proceeding is explained in detail. Using the described superposition $z^j := z_J^j - z_\lambda^j$, system (5.30) is reduced to

$$(u_s^{j\prime *}(J_u^{j\prime}) - u_s^{j\prime *}[\tau_{j+1}](\lambda^{j+1}), \delta s) = -(e_s^{j\prime *}(z^j), \delta s), \tag{5.38a}$$
$$\langle u_q^{j\prime *}(J_u^{j\prime}) - u_q^{j\prime *}[\tau_{j+1}](\lambda^{j+1}), \delta q\rangle_{Q^{j*}\times Q^j} = -\langle e_q^{j\prime *}(z^j), \delta q\rangle_{Q^{j*}\times Q^j}, \tag{5.38b}$$

where the right-hand side terms contain the adjoint solutions. Since the right-hand sides of the second equation of (5.29) and of (5.34) coincide, the following equalities are obtained using the weak form $e_s^{j\prime}(\delta s)(\varphi) := (e_s^{j\prime}(\delta s), \varphi)$:

$$(\delta s, e_s^{j\prime *}(z^j)) = (e_s^{j\prime}(\delta s), z^j) = e_s^{j\prime}(\delta s)(z^j) = (\delta s, z^j(\tau_j)).$$

Thus, replacing the adjoint terms in (5.27c*) by the corresponding term in (5.38a) for all $j \in \{0, \dots, M-1\}$ and using the last equality results in

$$\mathcal{L}'_{s^j}(\delta s) = (\lambda^j, \delta s) - (z^j(\tau_j), \delta s).$$

This is exactly the adjoint continuity condition (5.19c). Analogously, equation (5.38b) can be exploited to get (5.19b) and (5.19e). The right-hand sides of the first equation of (5.29) and of (5.33) coincide, which leaves us with

$$\langle e_q^{j\prime *}(z^j), \delta q\rangle_{Q^{j*}\times Q^j} = \langle e_q^{j\prime}(\delta q), z^j\rangle_{X^{j*}\times X^j} = e_q^{j\prime}(\delta q)(z^j) = -b'_q(q^j)(\delta q, z^j).$$

Here, we utilize the definition $\langle e_q^{j\prime}(\delta q), \varphi\rangle_{X^{j*}\times X^j} := e_q^{j\prime}(\delta q)(\varphi)$. Now, the adjoint terms in (5.27e*) can be replaced by the corresponding term in (5.38b). Then, use of the last equality for all $j \in \{0, \dots, M-1\}$ results in

$$\mathcal{L}'_{q^j}(\delta q) = \langle J_q^{j\prime}, \delta q\rangle_{Q^{j*}\times Q^j} + b'_q(q^j)(\delta q, z^j).$$

Since the last equation is identical to (5.19e) (and, for $j = 0$, to (5.19b)), this completes the proof. □

5.4 Summary and further aspects of IMS and DMS

The presentation of the IMS and DMS methods in Sections 5.1 and 5.2 shows that the two shooting approaches are closely related. Both methods solve the same system of equations, namely the extended KKT conditions (3.48). The affinity of both methods is underlined by the almost identical structure of Algorithms 5.1 and 5.4. Usually, DMS is used in a variant that was denoted as classical in Section 5.3. This classical DMS lacks an explicit occurrence of adjoint equations or adjoint continuity conditions, which obscures the relations between DMS and IMS. In the last section, the equivalence between this classical DMS approach and a new one, presented in Section 5.2, has been proved. From the non-classical DMS method, the relation to IMS can be directly deduced (see Figure 5.2). The differences between the approaches result from the different ways of splitting system (3.50). The splittings (5.1) and (5.17) induce different internal dependencies of the arguments of the common starting point, the extended OCP (3.43) – (3.44).

Remark 5.7. There are further possibilities of splitting the set of arguments of the extended Lagrangian (3.47). This could be a topic of further research, although it is not clear whether these splittings result in actually executable algorithms.

Subsections 5.1.2 and 5.2.2, where the main algorithmic steps of the shooting methods are discussed, focus on the solution of the shooting systems (5.4) resp. (5.20) by Newton's method. This may a priori lead to the impression that DMS is more expensive, because the shooting system to be solved comprises the temporally and spatially distributed discretized controls. However, we have seen that both IMS and DMS can be regarded as two-step fixed point iterations, and the solution of the shooting systems constitutes only the second step of the respective two-step method. If the shooting approaches are judged only by this criterion, IMS is unsurprisingly superior.

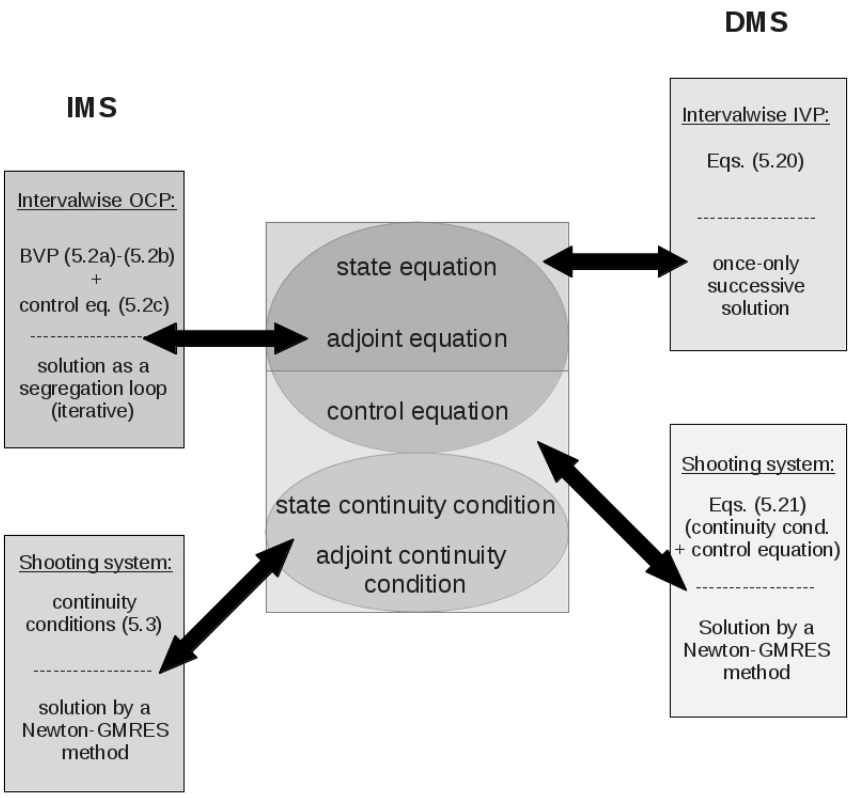

Figure 5.2. Illustration of the relationship between indirect and direct multiple shooting.

To obtain a complete picture, the whole shooting process in both approaches must be taken into account. In the DMS framework, the first part of the two-step fixed point iteration consists in solving a (nonlinear) IVP (5.18) on each subinterval, and the additional sensitivity equations are intervalwise (linear) IVP. The solution of IVP is straightforward, and a detailed description in Subsection 5.2.2 was skipped. In contrast, the corresponding IMS counterpart requires the solution of (nonlinear) subinterval BVP (5.2) that are smaller

versions of the original OCP. These intervalwise OCP are iteratively solved by a Newton-CG method, which makes the first step in IMS more expensive than its DMS equivalent. Moreover, the corresponding (linear) sensitivity problems (5.11) and (5.12) for the Newton-GMRES method, which constitute linear BVP, necessitate greater effort than their DMS counterpart. Summarizing, in IMS the first solution step is more expensive than in DMS, whereas for the second step of the respective two-step fixed-point methods the contrary holds true. Altogether, it is so far not clear whether one of the two approaches (IMS or DMS) should be preferred. The following abstract setting exemplifies the issues discussed so far.

Comparative example. Assume a distributed objective functional that is to be minimized subject to a linear parabolic PDE. Linearity ensures the solvability of all involved Newton methods within one Newton step. The solution interval is decomposed into $M = 10$ shooting intervals each discretized with $K = 100$ timesteps, and the spatial mesh comprises $N = 1000$ degrees of freedom. We further assume that all iterative methods in the solution process, such as Newton-CG or Newton-GMRES, need five iterations (which is, for simple examples, a realistic estimate). Table 5.1 below displays a contrasting juxtaposition of the numerical effort of both methods, concerning the number of BVP/IVP to be solved on the one hand and the size of the respective shooting system on the other hand.

Table 5.1. Comparative example: numerical effort for IMS and DMS in the framework of a simple linear-quadratic parabolic OCP.

	IMS	DMS
first step	solving interval OCP yields (twice): 10 state eqs., 10 adjoint eqs. 5 Newton-CG iterations each yield: 10 tangent eqs., 10 extra adjoint eqs. in total: 240 linear problems	solving interval IVP yields (twice): 10 state eqs., 10 adjoint eqs. in total: 40 linear problems
second step	system size: $2 \cdot M \cdot N = 20000$ (shooting conditions) matrix-free: 5 Newton-GMRES steps each yield: 2 linear BVP, update of length 20000	system size: $(K+2) \cdot M \cdot N = 1020000$ (shooting conditions + controls) matrix-free: 5 Newton-GMRES steps each yield: 5 linear IVP, update of length 1020000

Both IMS and DMS still comprise a variety of algorithms, depending on how the subinterval problems are solved (reduced approach as above vs. all-at-once approach), on how Newton's system is solved (matrix-free solver as above vs. explicit matrix assembling, inclusion of globalization techniques or of an SQP-like inexact Newton method), on how the sensitivity equations are solved (fixed-point method as above or more sophisticated approaches) etc. In order to judge both shooting approaches justly, more numerical tests have to be carried out,

taking into account the mentioned algorithmic variations and comparing their performance for a variety of different problems.

There is one more topic that has not been mentioned so far. In the optimal control literature, one frequently encounters a distinction between feasible and infeasible methods. We first state what is meant by feasibility (see also the textbooks by Geiger & Kanzow [43] or Nocedal & Wright [90]).

Definition 5.1. *A pair of functions (q, u) is called feasible for an OCP of type* (3.3) – (3.4) *if and only if it fulfils the PDE side condition including boundary and initial values. Analogously, pairs of intervalwise functions $(q^j, u^j)_{j=0}^{M-1}$ are called feasible for the OCP* (3.43) – (3.44) *if and only if they fulfil both the intervalwise PDE side condition and the continuity conditions.*

Note that this definition generalizes the notion of a feasible point (known from finite-dimensional optimization theory), but does not clarify what a feasible method is. As all methods for OCP usually involve iterative sub-algorithms, in solving an OCP we always end up with an approximating sequence $\{(q_k, u_k)\}_{k\in\mathbb{N}}$ or $\{(q_k^j, u_k^j)_{j=0}^{M-1}\}_{k\in\mathbb{N}}$. This leads to the following definition.

Definition 5.2. *An iterative method for solving OCP is called feasible if and only if the iterates (q_k, u_k) resp. $(q_k^j, u_k^j)_{j=0}^{M-1}$ follow a so-called feasible path, i. e., each single iterate is a feasible point (and thus fulfils all the side conditions of the OCP). Methods that do not necessarily yield feasible iterates are called infeasible.*

Judged by this criterion, both IMS and DMS constitute infeasible methods for the OCP (3.43) – (3.44). For DMS, a sequence of iterates $\{(q_k, u_k)\}_{k\in\mathbb{N}}$ is generated for any given set (s, q, λ) of initial shooting variables. However, only the last iterate fulfils the continuity conditions with sufficient accuracy and can therefore be interpreted as feasible. The same is true for IMS, but here at least the local subinterval solutions are always feasible points for the local OCP. They constitute correct solutions to subinterval OCP which are parameterized with boundary values that are ill-chosen with respect to the global problem. Thus, IMS is infeasible for the global OCP (3.43) – (3.44) but produces feasible partial solutions on all shooting intervals.

5.5 Numerical tests

In this section, the theoretical discussions and results from Sections 5.1 to 5.4 are illustrated by numerical examples. The framework of these examples has been presented at the end of Section 3.1. Subsection 5.5.1 starts with results for a linear and a nonlinear example in the IMS framework. We observe the stabilizing effect of multiple shooting. In addition, the symmetric Gauss-Seidel preconditioner from Subsection 5.1.2 is applied, and a comparison of results achieved both with and without the preconditioner justifies why we abstain from using it any further. Subsection 5.5.2 focusses on the DMS method presented in Section

5.2. However, as this work is not concerned with numerical linear algebra, and after the experience in the IMS case, the use of a preconditioner is skipped from the beginning, as it is harder to implement than its IMS counterpart. Further results for additional examples are presented in Subsection 5.5.3. Finally, in Subsection 5.5.4, some results are summarized that motivate our examination of adaptive shooting processes in Chapter 7. Similar results have been presented in Example 2.1 in the ODE framework. As holds for all PDE examples in this thesis, the computations in the current section have been carried out using the finite element software `deal.ii`; they rely upon the discretization routines presented in Section 4.1. Furthermore, as we often compare different approaches in terms of computing time, it is important to note that all computations were carried out on the same computer.

5.5.1 Results for IMS

Linear example. The first test example is a linear-quadratic OCP where the side condition depends on a parameter ω. For certain concrete choices of this parameter, the problem becomes unstable, which prevents the use of simple shooting and makes a splitting into several shooting intervals necessary. Several aspects that have been theoretically discussed in previous sections can be illustrated by this simple linear framework; our results highlight the stabilizing effect of multiple shooting methods including the examination of the symmetric Gauss-seidel preconditioner. Furthermore, this example is used to raise the question of a reasonable choice of the shooting point distribution (3.42).

Example 5.1. *Consider the following linear-quadratic OCP, which has already been treated by Hesse & Kanschat [53] under different aspects:*

$$\min_{(q,u)} \; J(q,u) = \frac{1}{2}\|u(x,T) - \hat{u}(x,T)\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\int_0^T \|q(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t,$$

subject to the nonstationary Helmholtz equation

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) &= q(x,t) && \text{in } \Omega \times (0,T], \\ u(x,t) &= 0 && \text{on } \partial\Omega \times [0,T], \\ u(x,0) &= \cos\Big(\frac{\pi}{2}x_1\Big)\cos\Big(\frac{\pi}{2}x_2\Big) && \text{in } \Omega. \end{aligned}$$

The computational domain is $\Omega = (-1,1)^2$ (the variable x always stands for (x_1, x_2)), the final time $T = 5$, and the regularization parameter $\alpha = 10^{-2}$. The Helmholtz parameter (reaction rate) ω runs through a set of integer values, usually $3 \le \omega \le 10$. In our setting, the initial value $u_0(x)$ is the eigenfunction corresponding to the smallest frequency of the Laplacian on the domain Ω, and the associated eigenvalue is $\pi^2/2 \approx 4.9348$. Note that both eigenvalues and -functions depend on the domain Ω. The goal is to fit the constant function $\hat{u}(x,5) \equiv 0.5$ at the final time $T = 5$. In Figure 5.3, we see (for $\omega = 7$) that in this case the state variable obviously matches this prescribed value at the final time, but develops a boundary layer due to the homogeneous Dirichlet boundary data that are

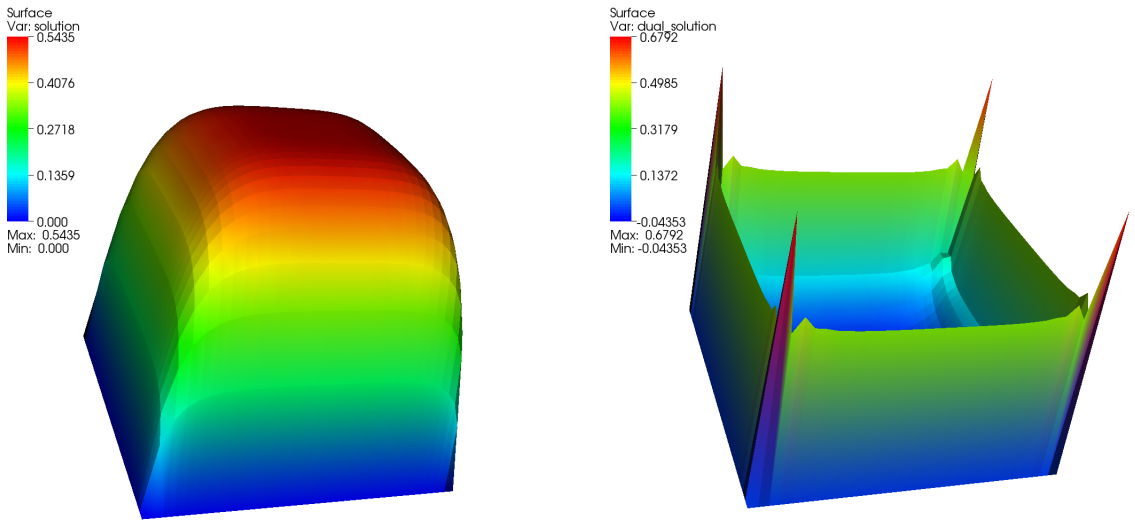

Figure 5.3. Example 5.1: State variable (left) and adjoint variable (right) for $\omega = 7$ at end time $T = 5$.

not compatible with the constant tracking function. The adjoint solution resembles a regularized line Dirac function along $\partial\Omega$.

The following results are obtained on a four times globally refined spatial mesh of 256 cells and with 500 uniform time steps, equally distributed to the shooting intervals in case of multiple shooting. The stopping criterion for the shooting process is reached if the size of the shooting residual falls below the tolerance value $\text{TOL} = 1.0 \cdot 10^{-05}$.
The example illustrates the benefit of multiple shooting in order to overcome possible instabilities in the problem. In fact, for values of ω that exceed the smallest eigenvalue of $-\Delta$, instabilities are expected to occur in the state equation. Consequently, the behavior of the solution algorithm on only one shooting interval (so-called indirect simple shooting, abbreviated by ISS) deteriorates at about $\omega = 5$. This effect is illustrated in Table 5.2, where simple shooting is compared to a state-of-the-art solution algorithm for parabolic optimization problems described by Becker et al. [7]. The latter method (denoted by SotA) solves the problem directly, i. e., without any time domain decomposition, by employing a Newton-CG algorithm, whereas simple shooting treats the problem as a BVP and uses Newton's method to solve the shooting system (5.4). The comparison is carried out with respect to the number of Newton-CG or Newton-GMRES steps needed for achieving the same accuracy in the optimal value of $J(q, u)$, and with respect to computing time. Table 5.2 reveals that ISS as well as the SotA algorithm both break down if the parameter ω is increased beyond the threshold $\omega = 5$. Furthermore we observe that simple shooting is more expensive than the shooting-free alternative method in terms of computing time. This is due to the matching conditions that have to be solved in addition.
In order to confirm that indirect simple shooting breaks down due to lacking stability in the state equation, the problem is solved for different end times $T \in \{1, 2, 3, 4, 5, 6\}$. As a result, for $\omega \in \{6, 7, 8\}$ indirect simple shooting is not able to integrate over long time

Table 5.2. Example 5.1: Comparison of a state-of-the-art algorithm (SotA) and indirect simple shooting (ISS) for different values of ω.

	SotA			ISS			
ω	#CG	$J(q,u)$	t(s)	#GMRES	$J(q,u)$	$\|F\|$	t(s)
3	10	0.0938	41	20	0.0938	$9.2{\cdot}10^{-12}$	145
4	10	0.0863	42	20	0.0863	$3.4{\cdot}10^{-11}$	149
5	12	0.0794	48	20	0.0794	$8.4{\cdot}10^{-10}$	151
6	–	–	–	22	0.0884	$2.5{\cdot}10^{-06}$	165
7	–	–	–	–	–	–	–

intervals. The results in Table 5.3 show that the time interval for the solution decreases with increasing values of ω. For the computations underlying Table 5.3, we used $100 \cdot T$ time steps.

Table 5.3. Example 5.1: Time integration with indirect simple shooting for varying ω and time intervals of increasing length.

T	$\omega = 5$		$\omega = 6$		$\omega = 7$		$\omega = 8$	
	$\#_{\text{GMRES}}$	$J(q,u)$	$\#_{\text{GMRES}}$	$J(q,u)$	$\#_{\text{GMRES}}$	$J(q,u)$	$\#_{\text{GMRES}}$	$J(q,u)$
1	20	0.0795	21	0.0842	22	0.0935	22	0.1037
2	20	0.0794	21	0.0867	22	0.0967	22	0.1057
3	20	0.0794	21	0.0878	22	0.0971	–	–
4	20	0.0794	22	0.0883	–	–	–	–
5	20	0.0794	22	0.0884	–	–	–	–
6	20	0.0794	–	–	–	–	–	–

As provided, indirect simple shooting becomes unstable if the time interval is too long or if the parameter ω becomes too large. Therefore, we focus on the multiple shooting algorithm from Section 5.1. If a fixed number of shooting intervals is chosen, there supposedly exists a new threshold for the parameter ω beyond which the computations break down. This supposition is used to compare the IMS algorithm without preconditioner to a modified IMS algorithm that was described at the end of Section 5.1 and involves a symmetric Gauss-Seidel (SGS) preconditioner. In Table 5.4, the results for five equidistant shooting intervals are presented. For $\omega \leq 5$, IMS yields equally good results as the SotA and ISS algorithms but takes more time (compare Table 5.2). However, while SotA and ISS fail for $\omega > 5$, the IMS method still works if ω is further increased. The unpreconditioned IMS works up to $\omega = 11$ (where two outer Newton-type iterations are needed), while the IMS method with SGS preconditioning already fails for $\omega = 9$. For $\omega \geq 12$ without preconditioner, respectively $\omega \geq 9$, using five shooting intervals is no longer sufficient for solving the problem. Although the number of GMRES iterations is reduced by up to 50%, we observe another disadvantage of the preconditioner: the computing time is significantly larger than for the unpreconditioned IMS algorithm. As provided in Section 5.1, the SGS

preconditioner only pays off if the number of GMRES iterations can be reduced by at least two thirds. This is not supported by our example.

Table 5.4. Example 5.1: Indirect multiple shooting on 5 shooting intervals (IMS_5) with and without SGS preconditioner for different values of ω.

	with preconditioner				without preconditioner			
ω	#GMRES	$J(q,u)$	$\|F\|$	t(s)	#GMRES	$J(q,u)$	$\|F\|$	t(s)
3	22	0.0938	$5.5{\cdot}10^{-11}$	450	25	0.0938	$3.5{\cdot}10^{-11}$	189
4	22	0.0863	$6.7{\cdot}10^{-11}$	437	28	0.0863	$9.5{\cdot}10^{-11}$	208
5	25	0.0794	$4.3{\cdot}10^{-10}$	497	43	0.0794	$1.7{\cdot}10^{-11}$	302
6	25	0.0884	$1.1{\cdot}10^{-09}$	499	43	0.0884	$5.1{\cdot}10^{-11}$	307
7	26	0.0972	$4.6{\cdot}10^{-09}$	527	44	0.0972	$3.1{\cdot}10^{-10}$	315
8	27	0.1058	$1.1{\cdot}10^{-06}$	1014	45	0.1058	$1.5{\cdot}10^{-09}$	321
9	–	–	–	–	46	0.1142	$7.1{\cdot}10^{-09}$	321
10	–	–	–	–	48	0.1225	$5.9{\cdot}10^{-08}$	338
11	–	–	–	–	51+50	0.1307	$3.1{\cdot}10^{-07}$	669
12	–	–	–	–	–	–	–	–

However, the advantage of a preconditioner might only become observable for larger systems. In fact, Table 5.5 shows some results in this regard. We therefore split the solution interval into different numbers of shooting intervals (SI) and observe that, if this decomposition is too fine, the unpreconditioned IMS does not work, while IMS with the SGS preconditioner still yields results. However, we maintain that it is advisable not to use the preconditioner in order to save computing time for a small number of shooting intervals.

Table 5.5. Example 5.1: IMS with and without SGS preconditioner for $\omega = 7$ (where $J(q,u) = 0.0972$) and different equidistant shooting decompositions.

	with preconditioner			without preconditioner		
#SI	#GMRES	$\|F\|$	t(s)	#GMRES	$\|F\|$	t(s)
2	23	$1.4{\cdot}10^{-07}$	457	25	$4.1{\cdot}10^{-06}$	185
5	26	$4.6{\cdot}10^{-09}$	527	44	$3.1{\cdot}10^{-10}$	315
10	45	$3.6{\cdot}10^{-11}$	888	108	$2.7{\cdot}10^{-11}$	702
20	111	$1.8{\cdot}10^{-11}$	2120	–	–	–

Another important aspect resulting from Table 5.5 is the growing number of GMRES iterations in the preconditioned case. Hesse [52] claims that with SGS preconditioning, the number of GMRES iterations required in the solution process remains constant for increasing shooting systems. She draws this conclusion from only one example. Such a property is not observed by Comas [26] or Heinkenschloss [50], who were the first to apply SGS preconditioners in the multiple shooting context. Table 5.6 presents the attempt to replicate Hesse's results. As diplayed, we are not able to reproduce her findings. On the

contrary the table illustrates that, even with SGS preconditioning, the number of GMRES iterations increases with the system size. Furthermore, in this example there is almost no reduction in the number of GMRES iterations, and therefore the SGS preconditioned IMS method performs worse than the unpreconditioned one with respect to computing time.

Table 5.6. Example 5.1: IMS with and without SGS preconditioner for $\omega = 0$ (the heat equation) and different equidistant shooting decompositions.

	with preconditioner			without preconditioner		
#SI	#GMRES	$\|F\|$	t(s)	#GMRES	$\|F\|$	t(s)
5	21	$3.5{\cdot}10^{-07}$	494	21	$3.5{\cdot}10^{-07}$	189
10	23	$4.0{\cdot}10^{-06}$	551	24	$4.0{\cdot}10^{-06}$	209
15	27	$8.8{\cdot}10^{-06}$	628	35	$8.8{\cdot}10^{-06}$	293
20	36	$3.8{\cdot}10^{-06}$	837	38	$3.8{\cdot}10^{-06}$	315
25	41	$6.0{\cdot}10^{-06}$	944	46	$6.0{\cdot}10^{-06}$	373
30	45	$8.0{\cdot}10^{-06}$	1068	49	$8.0{\cdot}10^{-06}$	396

Nonlinear example. We now extend the first example by introducing the additional nonlinear reaction term u^3 to the constraining Helmholtz equation. Furthermore, our goal in Example 5.2 below is to match a function $\hat{u}(x,t)$ on the whole time interval, i. e., the functional is now of distributed tracking type.

Example 5.2. *We consider the problem*

$$\min_{(q,u)} \; J(q,u) = \frac{1}{2}\int_0^T \|u(x,t) - \hat{u}(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t + \frac{\alpha}{2}\int_0^T \|q(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t,$$

subject to the nonstationary nonlinear Helmholtz problem

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + u^3(x,t) &= q(x,t) \quad \text{in } \Omega \times (0,T], \\ u(x,t) &= 0 \quad \text{on } \partial\Omega \times [0,T], \\ u(x,0) &= u_0(x) \quad \text{in } \Omega. \end{aligned}$$

The computational domain is $\Omega = (-1,1)^2$, and the end time is $T = 5$. We fix the regularization parameter $\alpha = 0.5$ and the Helmholtz parameter $\omega = 7$ at a value for which simple shooting is expected to fail. Furthermore, the tracking function is chosen as

$$\hat{u}(x,t) := \begin{cases} \frac{2}{5}t \cdot \left(1 - x_1^{12}\right)\left(1 - x_2^{12}\right), & t \le \frac{5}{2}, \\ \left(\frac{2}{5}t - 2\right) \cdot \left(1 - x_1^{12}\right)\left(1 - x_2^{12}\right), & t > \frac{5}{2}, \end{cases}$$

with zero boundary conditions and a maximum absolute value at the center $(0,0)$ of Ω. The function $\hat{u}(x,t)$ evolves linearly in time and has a jump at the midpoint of the time interval. The initial function $u_0(x) \equiv 0$ is chosen such that it fits the value $\hat{u}(x,0)$. Our

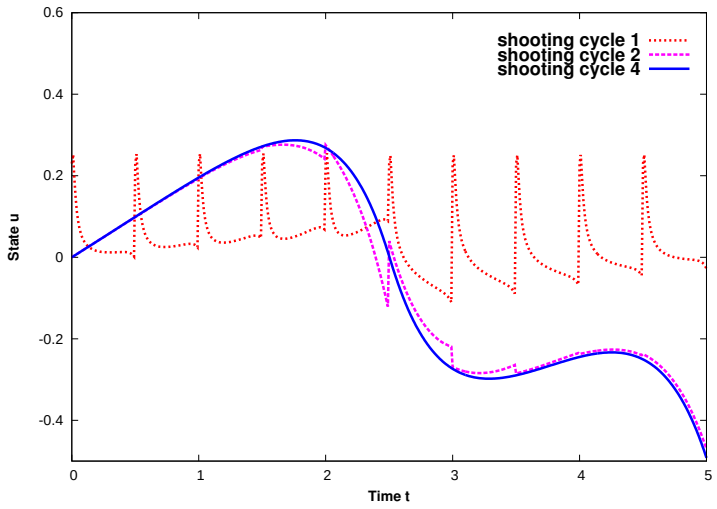

Figure 5.4. Example 5.2: The state $u(0,0,t)$ at different IMS cycles.

Table 5.7. Example 5.2: IMS with and without SGS preconditioner; the computing time was 12300 sec. with and 7438 sec. without preconditioning.

	with preconditioner			without preconditioner		
Newton it.	#GMRES	$\|F\|$	$J(q,u)$	#GMRES	$\|F\|$	$J(q,u)$
0	–	$4.0{\cdot}10^{-00}$	2.262	–	$4.0{\cdot}10^{-00}$	2.262
1	12	$2.2{\cdot}10^{-01}$	2.301	25	$2.2{\cdot}10^{-01}$	2.301
2	11	$4.3{\cdot}10^{-03}$	2.179	24	$4.4{\cdot}10^{-03}$	2.179
3	10	$2.0{\cdot}10^{-05}$	2.180	24	$1.4{\cdot}10^{-05}$	2.180

computations are again carried out on the same space mesh, but this time we choose 10 equally distributed shooting intervals each of which comprises 50 interior time steps. We stop the computation as soon as the shooting residual $\|F\|$ (where F is given by (5.3)) becomes smaller than TOL $= 1.0 \cdot 10^{-03}$. Figure 5.4 shows the temporal development of the state variable $u(0,0,t)$ by evaluating the solution at the center of the spatial domain in different cycles of the multiple shooting procedure. The corresponding controls are displayed in Figure 5.5. In the first iteration with arbitrary initial values (dotted curves), we can clearly distinguish the 10 shooting intervals. Jumps are visualized by vertical lines. The second shooting cycle (dashed curves) is already close to convergence, but more shooting cycles are needed to reach the prescribed tolerance (solid curves). In Table 5.7 we present the development of the solution and confirm that the SGS preconditioner is not efficient for multiple shooting algorithms.

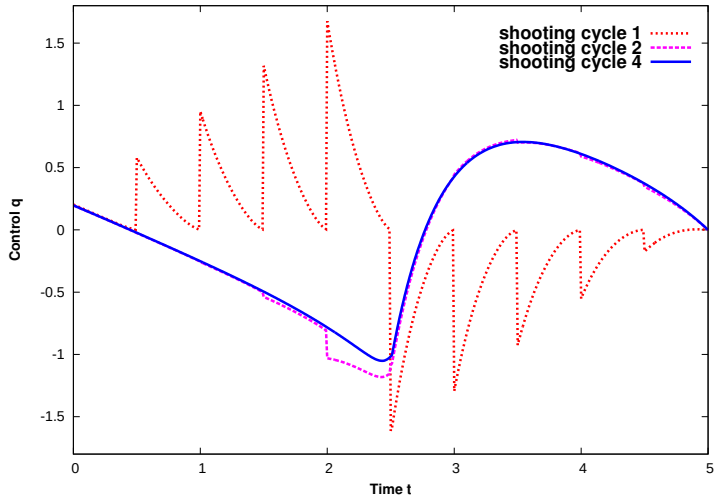

Figure 5.5. Example 5.2: The control $q(0, 0, t)$ at different IMS cycles.

5.5.2 Results for DMS

Linear example. The results presented in this subsection were achieved with the DMS method from Section 5.2. For the sake of comparability, the same two problems are considered as in Subsection 5.5.1, i. e., a linear and a nonlinear example.
First, Example 5.1 is reconsidered with the same domain, discretization and tolerances. Again, the aim is to detect the threshold for the Helmholtz parameter, $\omega \approx 5$, in the direct simple shooting (DSS) framework. This unstable behavior necessitates multiple shooting, and we consider again five equidistantly distributed shooting intervals.

Table 5.8. Example 5.1: Direct simple (DSS) and multiple shooting (DMS_5) for different values of ω.

	DSS				DMS_5			
ω	#GMRES	$J(q,u)$	$\|F\|$	t(s)	#GMRES	$J(q,u)$	$\|F\|$	t(s)
3	54	0.0938	$3.4{\cdot}10^{-10}$	303	62	0.0938	$3.2{\cdot}10^{-10}$	367
4	56	0.0863	$5.6{\cdot}10^{-10}$	322	70	0.0863	$3.5{\cdot}10^{-10}$	405
5	56	0.0794	$1.3{\cdot}10^{-09}$	324	98	0.0794	$3.5{\cdot}10^{-10}$	556
6	56	0.0884	$2.1{\cdot}10^{-08}$	324	102	0.0884	$4.1{\cdot}10^{-10}$	586
7	–	–	–	–	102	0.0972	$9.0{\cdot}10^{-10}$	585
8	–	–	–	–	106	0.1058	$1.7{\cdot}10^{-09}$	606
9	–	–	–	–	110	0.1142	$3.4{\cdot}10^{-09}$	631
10	–	–	–	–	112	0.1225	$1.1{\cdot}10^{-08}$	652
11	–	–	–	–	–	–	–	–

Table 5.8 shows the corresponding results. The DSS block on the left should be compared to the right block of Table 5.2, whereas the DMS_5 block on the right corresponds to the

right block of Table 5.4. In both cases direct shooting is more expensive than indirect shooting in terms of computing time.

Nonlinear example. For the nonlinear example, we revisit Example 5.2. The aim is to confirm that the state and control solutions coincide with those from the IMS approach after convergence, despite the control being computed in a different way as part of the shooting system. In Figure 5.6, the temporal evolution of the DMS state $u(0,0,t)$ at the center of the spatial domain Ω is presented. This corresponds to Figure 5.4. Comparing the solid lines, i. e., the state solution after multiple shooting is converged, the results of IMS and DMS coincide.

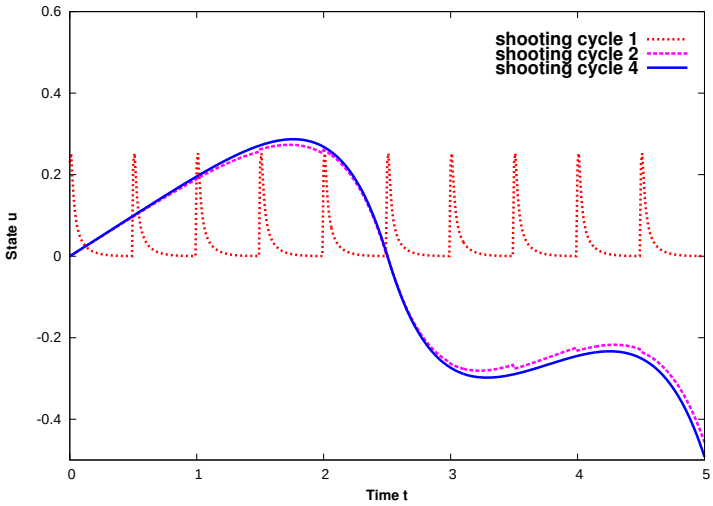

Figure 5.6. Example 5.2: The state $u(0,0,t)$ at different DMS cycles.

Comparing the results of Table 5.9, achieved on a shooting grid of 10 equidistant shooting intervals, to those displayed in Table 5.7 shows that IMS and DMS finally yield the same functional value despite their different feasibility properties (see Section 5.4).

Table 5.9. Example 5.2: DMS for a nonlinear example; the computing time was 2437 sec.

Newton it.	#GMRES	$\|F\|$	$J(q,u)$
0	–	$2.1{\cdot}10^{+01}$	2.261
1	53	$5.1{\cdot}10^{-01}$	2.194
2	53	$5.4{\cdot}10^{-03}$	2.181
3	53	$1.0{\cdot}10^{-06}$	2.180

As a final result of this subsection, we briefly discuss how the results depend on the regularization parameter α. Although the comparison of Figures 5.4 and 5.6 shows that both shooting approaches yield equally good results, the actual quality of these results is dubitable when compared to the tracking function $\hat{u}$. The temporal development of $\hat{u}$ at

the origin $(0,0)$ of the spatial domain Ω is a piecewise linear function with a jump of height 2 at $t = 2.5$ which first connects the points $(t, \hat{u}(0,0,t)) = (0,0)$ and $(t, \hat{u}(0,0,t)) = (2.5,1)$ and then the points $(t, \hat{u}(0,0,t)) = (2.5,-1)$ and $(t, \hat{u}(0,0,t)) = (0,0)$. The state solution u as in Figures 5.4 and 5.6 is a bad match for $\hat{u}(0,0,t)$. A first major deviation is the range of function values, which is about $[-0.48; 0.36]$ rather than $[-1,1]$; the second mismatch is the behavior of $u(0,0,t)$ towards the end of the interval, where the values decrease further instead of approaching the correct value $\hat{u}(0,0,5) = 0$.

Table 5.10. Example 5.2: Dependence of the functional value on the regularization parameter α.

α	#Newton	#GMRES$_{\text{min / max}}$	$\|F\|$	$J(q,u)$	t(s)
1	3	53/53	$3.5{\cdot}10^{-07}$	2.354	2466
0.5	3	53/53	$1.0{\cdot}10^{-06}$	2.180	2437
0.1	4	72/76	$3.3{\cdot}10^{-05}$	1.478	4559
0.05	4	80/83	$3.8{\cdot}10^{-04}$	1.172	4979
0.01	5	103/106	$4.2{\cdot}10^{-07}$	0.718	8012

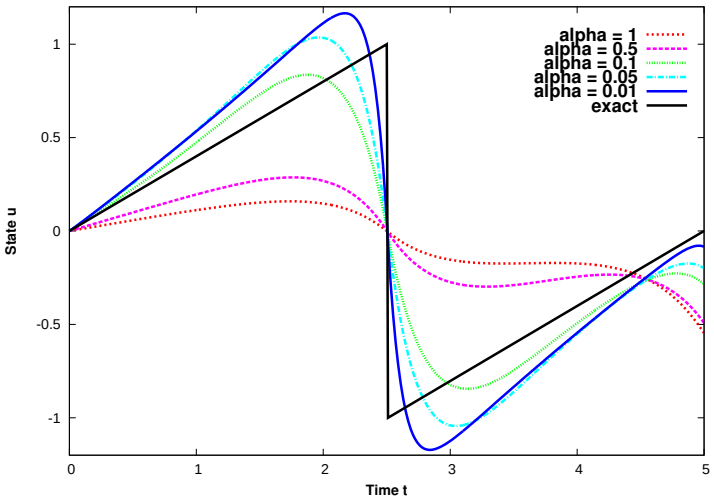

Figure 5.7. Example 5.2: Quality of the match depending on the regularization parameter α.

Closing this subsection, we confirm that these major deviations are due to our large choice $\alpha = 0.5$ of the regularization parameter. Better matches for $\hat{u}$ are expected if the influence of the cost term is reduced by choosing smaller values of α. Table 5.10 and Figure 5.7 confirm this: the smaller we choose α, the smaller the functional value becomes after convergence. Figure 5.7 displays the corresponding evolution curves of $u(0,0,t)$; the improvement of the matching with decreasing α is illustrated.

5.5.3 Comparison of IMS and DMS

In the two previous subsections, examples for IMS and DMS were presented separately. We now consider further examples to draw a direct comparison between the two shooting approaches.

Linear example. The linear-quadratic OCP is considered on the rectangular space-time domain $\Omega \times I = \big([-1;3] \times [-1,1]\big) \times [0,1]$ and aims at matching a given state profile $\hat{u}(x,T)$ at the time interval endpoint $T = 1$.

Example 5.3. *We consider the problem*

$$\min_{(q,u)} \; J(q,u) = \frac{1}{2}\|u(x,T) - \hat{u}(x,T)\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\int_0^T \|q(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t, \tag{5.39}$$

subject to the nonstationary linear Helmholtz equation

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) &= q(x,t) && \text{in } \Omega \times (0,T], \\ u(x,t) &= 0 && \text{on } \partial\Omega \times [0,T], \\ u(x,0) &= \max\Big\{0, \cos\Big(\frac{\pi}{2}x_1\Big)\cos\Big(\frac{\pi}{2}x_2\Big)\Big\} && \text{on } \Omega. \end{aligned}$$

The profile to be tracked at $T = 1$ is chosen as $\hat{u}(x,1) = \min\Big\{0, \cos\Big(\frac{\pi}{2}x_1\Big)\cos\Big(\frac{\pi}{2}x_2\Big)\Big\}$. Thus, we expect the state solution to be a cosine bump moving from the left half to the right half of the spatial domain over time, thereby changing its sign.
We compute solutions of this problem for different values of the parameter ω and for $\alpha = 0.01$ by means of IMS and DMS. We use a four times globally refined spatial mesh (512 cells) and five equidistant shooting intervals, each discretized by 100 time steps. The results are shown in Table 5.11: The columns from left to right display the number of GMRES iterations, the functional value, the residual of the respective shooting system (which also serves as stopping criterion), and the computing time in seconds. As the problem is linear, we need only one Newton step for solving the system of shooting conditions.

Table 5.11. Example 5.3: Comparison of IMS and DMS for varying ω (required: $\|F\| < 5.0 \cdot 10^{-5}$) in a linear framework.

	IMS				DMS			
ω	$\#_{\text{GMRES}}$	$J(q,u)$	$\|F\|$	t(s)	$\#_{\text{GMRES}}$	$J(q,u)$	$\|F\|$	t(s)
0	52	0.0446	$1.6 \cdot 10^{-11}$	1497	110	0.0446	$1.9 \cdot 10^{-10}$	2507
1	64	0.0367	$1.8 \cdot 10^{-11}$	1825	128	0.0367	$2.2 \cdot 10^{-10}$	2909
2	76	0.0290	$2.1 \cdot 10^{-11}$	2149	156	0.0290	$2.3 \cdot 10^{-10}$	3531
3	83	0.0218	$2.4 \cdot 10^{-11}$	2347	192	0.0218	$2.3 \cdot 10^{-10}$	4360
4	130	0.0163	$2.6 \cdot 10^{-11}$	3601	248	0.0163	$2.6 \cdot 10^{-10}$	5586
5	165	0.0148	$2.9 \cdot 10^{-11}$	4571	416	0.0148	$2.8 \cdot 10^{-10}$	9423

Both methods have been implemented as described in Sections 5.1 and 5.2 and do not include additional tuning (such as condensing, reduction of control spaces etc.). Furthermore, they are used without preconditioning. For increasing ω the number of inner GMRES iterations grows in both cases, reflecting the deterioration of the conditioning of the respective problems. With DMS, more GMRES steps are required than with IMS, which is due to the larger linear shooting system entailing a higher condition number. The functional values $J(q,u)$ coincide for both methods, and the shooting residual $\|F\|$ is of comparable size. However, the DMS algorithm takes longer (by a factor of 1.5 up to 2) than IMS to solve the problem with the same accuracy. Finally, in Figure 5.8 we see that after convergence of IMS the expected wandering and inversion of the cosine bump is reproduced.

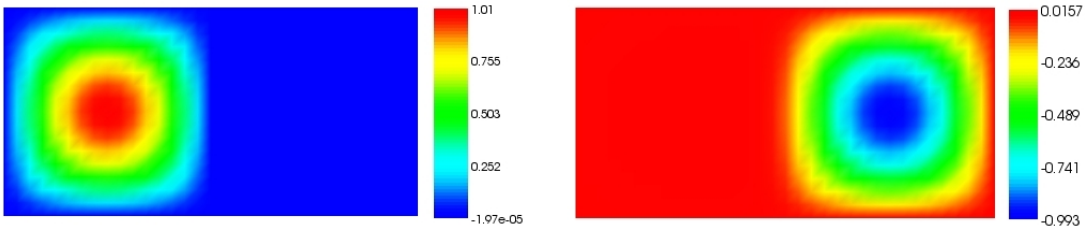

Figure 5.8. Example 5.3: Contour plot of the IMS solution on 5 shooting intervals after convergence: initial solution at time $T = 0$ (left), final solution at time $T = 1$ (right).

Nonlinear examples. The first nonlinear example is a modification of Example 5.2. In the following, the distributed tracking term is replaced by an end-time tracking term (with tracking function $\hat{u}_T \equiv 0.5$), the regularization parameter is fixed as $\alpha = 0.05$, and the initial condition is $u_0(x) = \cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)$, as in the linear Example 5.1.

Example 5.4. *Consider the problem*

$$\min_{(q,u)} J(q,u) = \frac{1}{2}\|u(x,T) - \hat{u}_T\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\int_0^T \|q(x,t)\|^2_{L^2(\Omega)}\,dt$$

on $\Omega = (-1,1)^2$ *and with end-time* $T = 5$*, subject to the nonstationary nonlinear Helmholtz equation*

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + u(x,t)^3 &= q(x,t) \quad \text{in } \Omega \times I, \\ u(x,t) &= 0 \quad \text{on } \partial\Omega \times I, \\ u(x,0) &= u_0(x) \quad \text{in } \Omega. \end{aligned}$$

Table 5.12 displays the results for this example on 10 shooting intervals and for varying values of the Helmholtz parameter ω. As the functional values for IMS and DMS coincide, they are presented only once. Analogously to Example 5.3 above, DMS requires more GMRES iterations per Newton step. Nevertheless, IMS is slower than DMS by a factor 2–3 in this nonlinear example.

Table 5.12. Example 5.4: Comparison of IMS (left) and DMS (right) on 10 intervals for varying values of ω (required: $\|F\| < 5.0 \cdot 10^{-05}$).

		IMS			DMS		
ω	$J(q,u)$	$\#_{\text{Newt(GMRES)}}$	$\|F\|$	$t(s)$	$\#_{\text{Newt(GMRES)}}$	$\|F\|$	$t(s)$
3	0.171	3 (26)	$8.0 \cdot 10^{-6}$	3243	3 (45)	$9.6 \cdot 10^{-9}$	1994
4	0.146	3 (42)	$1.2 \cdot 10^{-5}$	5410	3 (56)	$1.6 \cdot 10^{-6}$	2396
5	0.115	4 (44)	$1.9 \cdot 10^{-5}$	9492	5 (92)	$6.6 \cdot 10^{-10}$	4961
6	0.149	4 (53)	$1.5 \cdot 10^{-5}$	14876	4 (115)	$3.3 \cdot 10^{-6}$	6718
7	0.198	5 (54)	$4.9 \cdot 10^{-5}$	22367	4 (134)	$8.4 \cdot 10^{-8}$	7324

At the beginning of the shooting iteration, we are far from the solution, as the arbitrarily chosen starting values for the shooting variables lead to solution iterates that are not continuous. It is therefore not necessary to carry out the first shooting iterations on a fine spatial mesh. Instead, in order to save computational effort we can start on a coarse mesh, i. e., only 4 cells in Example 5.4, and carry out one Newton step to obtain an update for the shooting variables. The updated shooting variables are expected to better approximate the actual solution in the shooting points. Therefore, we refine the mesh globally, interpolate the updated shooting variables and carry out the next Newton step on the refined mesh which yields an even better approximation. This process is repeated until a maximally refined mesh is reached, i. e., the four times refined mesh with 256 cells, on which the remaining iterations until convergence are carried out. This global refinement process is illustrated in Figure 5.9.

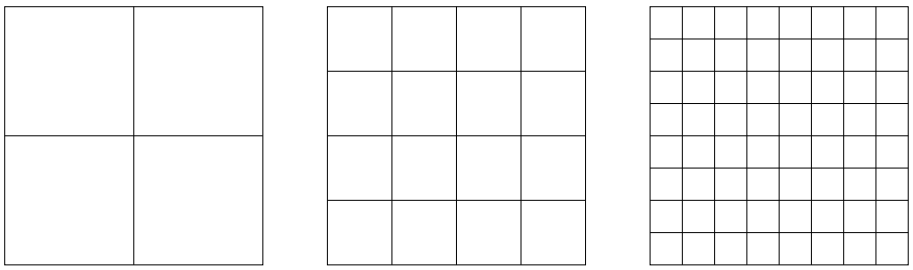

Figure 5.9. Example 5.4: Three consecutive globally refined meshes.

The results of IMS and DMS with the global refinement process included are displayed in Table 5.13. The number of GMRES iterations per Newton step varies due to the different spacial refinement levels. While on coarse meshes, few GMRES iterations are required, their number increases with mesh refinement, i. e., with increasing number of degrees of freedom in space and thus increasing size of the shooting system. For both IMS and DMS, a reduction in computing time of 20–50% is observed when compared to the results of Table 5.12 where the computations were carried out completely on the finest mesh.

Table 5.13. Example 5.4: Comparison of IMS (left) and DMS (right) on 10 intervals for varying values of ω (required: $\|F\| < 5.0 \cdot 10^{-05}$). Approximation of the shooting variables performed on successively refined grids. The respective third columns indicate the reduction of computing time in % compared to Table 5.12.

		IMS			DMS		
ω	$J(q,u)$	$\#_{\text{Newt(GMRES)}}$	$\|F\|$	%	$\#_{\text{Newt(GMRES)}}$	$\|F\|$	%
3	0.171	4 (19/26)	$1.5 \cdot 10^{-5}$	59.6	5 (21/45)	$1.2 \cdot 10^{-10}$	20.4
4	0.146	5 (20/28)	$2.3 \cdot 10^{-5}$	43.6	5 (21/53)	$4.2 \cdot 10^{-9}$	22.2
5	0.115	5 (20/44)	$5.6 \cdot 10^{-6}$	45.8	5 (21/73)	$1.3 \cdot 10^{-6}$	52.8
6	0.149	4 (26/53)	$2.0 \cdot 10^{-5}$	40.8	5 (56/112)	$3.2 \cdot 10^{-9}$	18.6
7	0.198	5 (20/54)	$3.5 \cdot 10^{-5}$	53.2	5 (21/112)	$2.9 \cdot 10^{-7}$	47.0

The second nonlinear problem is a modification of Example 5.3 consisting in the choice of a different regularization parameter $\alpha = 0.05$ and in an additional polynomial nonlinearity in the PDE side condition.

Example 5.5. *Consider the following problem: Minimize the functional* (5.39)*, subject to the semilinear Helmhotz-type equation*

$$\partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + u(x,t)^3 = q(x,t) \quad \text{in } \Omega \times (0,T].$$

The initial condition, the boundary values and the computational domain are chosen as in the configuration of Example 5.3.

Table 5.14. Example 5.5: Comparison of IMS and DMS for varying ω (required: $\|F\| < 1.0 \cdot 10^{-3}$) in a nonlinear framework.

	IMS				DMS			
ω	$\#_{\text{GMRES}}$	$J(q,u)$	$\|F\|$	t(s)	$\#_{\text{GMRES}}$	$J(q,u)$	$\|F\|$	t(s)
0	24/51	0.1639	$3.1 \cdot 10^{-6}$	2530	28/53	0.1639	$3.1 \cdot 10^{-5}$	2088
1	26/52	0.1420	$6.4 \cdot 10^{-6}$	2795	38/62	0.1420	$1.5 \cdot 10^{-5}$	2427
2	28/56	0.1187	$2.5 \cdot 10^{-6}$	3118	43/74	0.1187	$9.0 \cdot 10^{-5}$	2926
3	28/75	0.0948	$3.9 \cdot 10^{-6}$	4201	51/84	0.0948	$1.4 \cdot 10^{-4}$	3280
4	28/79	0.0735	$5.6 \cdot 10^{-6}$	4713	68/108	0.0735	$2.1 \cdot 10^{-4}$	4201
5	28/94	0.0645	$1.2 \cdot 10^{-5}$	5658	80/139	0.0645	$2.6 \cdot 10^{-4}$	5376

For all results presented in Table 5.14, i. e., IMS and DMS with an arbitrary parameter ω, four Newton iterations were required. Again, the global refinement strategy described for Example 5.4 was employed. Table 5.14 shows the results of this approach; the IMS and DMS methods provide equally good minimum functional values and shooting residuals and take roughly the same computing time.

Remark 5.8. The global refinement technique cannot be applied to linear examples, where the first Newton iteration already yields the converged solution. Thus, in the linear case we start on the most refined level. The computing times given in Tables 5.11 and 5.14 reflect this difference in the implementation. Even though the linear example takes more computing time than the corresponding nonlinear one, we emphasize that the direct comparison is not fair. If we included the global refinement process into the linear framework artificially, enforcing it to take several shooting iterations, then we could again benefit from the described refinement strategy.

5.5.4 Choice of the shooting intervals

Indirect shooting. The results in Table 5.4 raise the question of how many shooting intervals are at least needed, depending on the Helmholtz parameter ω, to solve the linear problem from Example 5.1 with only one outer Newton iteration. Table 5.15 gives a corresponding answer for values $5 \leq \omega \leq 10$ and illustrates that for increasing ω, the least number of shooting intervals also increases. In turn the problems get more and more ill-conditioned. For even larger values of ω, the problems are no longer solvable. Either the number of shooting intervals is too small, or the systems to be solved are too ill-conditioned. Therefore, employing a suitable preconditioner as discussed at the end of Subsection 5.1.2 becomes indispensable. Note, however, that our results were obtained

Table 5.15. Example 5.1: Minimum number of shooting intervals (SI) for IMS depending on ω.

ω	$\#_{\text{GMRES}}$	$\#_{\text{SI}}$	$J(q,u)$	$\|F\|$
5	20	1	0.0794	$3.5{\cdot}10^{-09}$
6	24	2	0.0884	$4.9{\cdot}10^{-07}$
7	26	3	0.0972	$2.7{\cdot}10^{-07}$
8	28	4	0.1058	$2.1{\cdot}10^{-06}$
9	48	5	0.1142	$9.1{\cdot}10^{-08}$
10	49	5	0.1225	$5.0{\cdot}10^{-07}$

without any preconditioning. Finally, from Table 5.16, we see that the minimum total number of shooting intervals that still yields a solution by performing one single outer step is the most efficient one. This results from an increase in computing time with a growing number of shooting intervals.

Direct shooting. We reconsider Example 5.1, for direct multiple shooting and our objective is again to find how many shooting intervals are needed to solve this linear problem with only one outer Newton iteration. The results are presented in Table 5.17 which states that, as in the IMS case above, an increase of ω leads to an increasing number of shooting intervals. Similarly, Table 5.18 confirms the results of Table 5.1 for the DMS case. We conclude that it is advisable to work with as few shooting intervals as possible to be maximally efficient.

Table 5.16. Example 5.1: Computing time for IMS depending on the number of shooting intervals (SI).

	$\omega = 7$				$\omega = 8$			
$\#_{\text{SI}}$	$\#_{\text{Newton}}$	$\#_{\text{GMRES}}$	t(s)	$\|F\|$	$\#_{\text{Newton}}$	$\#_{\text{GMRES}}$	t(s)	$\|F\|$
2	–	–	–	–	–	–	–	–
3	1	26	412	$2.7{\cdot}10^{-07}$	–	–	–	–
4	1	28	434	$3.0{\cdot}10^{-08}$	1	28	437	$2.1{\cdot}10^{-06}$
5	1	46	685	$2.2{\cdot}10^{-09}$	1	45	683	$1.6{\cdot}10^{-08}$
6	1	48	718	$1.1{\cdot}10^{-09}$	1	49	733	$5.7{\cdot}10^{-09}$
7	1	52	776	$6.8{\cdot}10^{-10}$	1	53	781	$2.5{\cdot}10^{-09}$
8	1	56	826	$4.2{\cdot}10^{-10}$	1	56	835	$1.5{\cdot}10^{-09}$
9	1	72	1061	$3.0{\cdot}10^{-10}$	1	79	1172	$9.3{\cdot}10^{-10}$
10	1	103	1492	$2.2{\cdot}10^{-10}$	1	109	1572	$6.5{\cdot}10^{-10}$

Table 5.17. Example 5.1: Minimum number of shooting intervals (SI) for DMS depending on ω.

ω	$\#_{\text{GMRES}}$	$\#_{\text{SI}}$	$J(q,u)$	$\|F\|$
5	50	1	0.0794	$4.3{\cdot}10^{-10}$
6	52	1	0.0884	$2.5{\cdot}10^{-08}$
7	56	2	0.0972	$2.2{\cdot}10^{-08}$
8	72	3	0.1057	$1.5{\cdot}10^{-08}$
9	82	4	0.1142	$1.1{\cdot}10^{-08}$
10	104	5	0.1225	$1.2{\cdot}10^{-08}$

From Tables 5.4, 5.8 or 5.11, the IMS method appears in general more efficient than its DMS counterpart when considering linear-quadratic OCP. This is further confirmed by the computing times given in Tables 5.16 and 5.18.

Interpretation of results. The numerical results from Tables 5.15 – 5.18, were achieved by a trial and error process. In order to avoid this time-consuming proceeding, criteria are desirable for adaptively determining the optimal total number as well as the optimal position of shooting points. This proper choice of shooting points τ_j is a critical issue especially in the PDE context, since with an increasing number of shooting points the dimension of the shooting system (5.4) resp. (5.20) grows ever larger. This deteriorates the conditioning of the shooting system, which in turn leads to a significant increase in computational effort. Therefore, an adaptive determination process for the shooting points is crucial for the efficient solution of problems that respond sensitively to perturbations in the data or modifications of certain parameter values.
However, even for ODE-governed BVP with solution $y(t;s)$, there are only few results concerning this question. Maier [80] developed a method that starts from a given shooting point distribution and automatically discards or inserts shooting points whenever necessary. The main drawback of his approach is its limitation to a certain problem class, namely

Table 5.18. Example 5.1: Computing time for DMS depending on the number of shooting intervals (SI)

	$\omega = 7$				$\omega = 8$			
$\#_{\text{SI}}$	$\#_{\text{Newton}}$	$\#_{\text{GMRES}}$	t(s)	$\|F\|$	$\#_{\text{Newton}}$	$\#_{\text{GMRES}}$	t(s)	$\|F\|$
2	1	56	672	$2.2{\cdot}10^{-08}$	–	–	–	–
3	1	70	840	$4.4{\cdot}10^{-09}$	1	72	866	$1.5{\cdot}10^{-08}$
4	1	80	970	$2.1{\cdot}10^{-09}$	1	82	969	$4.7{\cdot}10^{-09}$
5	1	100	1193	$1.5{\cdot}10^{-09}$	1	102	1211	$2.4{\cdot}10^{-09}$
6	1	106	1250	$1.2{\cdot}10^{-09}$	1	110	1297	$1.5{\cdot}10^{-09}$
7	1	132	1554	$9.6{\cdot}10^{-10}$	1	134	1565	$1.2{\cdot}10^{-09}$
8	1	164	1948	$8.0{\cdot}10^{-10}$	1	166	1963	$1.0{\cdot}10^{-09}$
9	1	214	2537	$6.8{\cdot}10^{-10}$	1	248	2933	$8.0{\cdot}10^{-10}$
10	1	298	3504	$6.0{\cdot}10^{-10}$	1	308	3621	$6.4{\cdot}10^{-10}$

singularly perturbed ODE BVP. Alternatively, based on prior work of Mattheij on the conditioning of linear BVP (see [81],[82]), Mattheij & Staarink [83] suggested to impose a bound for the growth of the sensitivities $G(t) := \frac{d}{ds}y(t;s)$ (see Subsection 2.2). Proceeding forward in time, whenever $\|G(t)\|$ exceeds a pre-chosen threshold value C ($\|\cdot\|$ being an arbitrary matrix norm), the current time-point t_i is taken as a new shooting point τ_j. This method has some major drawbacks which render the process rather heuristic. More importantly, the approach does not work for nonlinear problems, for in the nonlinear case $G(t) = G(t;s)$ and $\tau_j = \tau_j(s)$, where sensitivities and shooting points depend on the shooting variables s. This enforces a redistribution of the shooting points in each iteration of the Newton-type solver for the shooting conditions. Furthermore, the transfer of the method to the PDE context is neither clear in the linear case.
The necessity of matrix-free computation has been emphasized in Subsections 5.1.2 and 5.2.2, meaning that the sensitivity matrices are not available. Instead, we only have directional derivatives $u_s, u_\lambda, z_s, z_\lambda$ (solutions of the sensitivity problems (5.11) and (5.12) resp. (5.22) and (5.23)). Choosing a norm of the sensitivities as bounding constant C is thus not feasible in the PDE case. As many questions remain unanswered even for the linear ODE case, we did not present nonlinear examples here. Chapter 7 deals with these problems, and important questions on the subject of adaptivity are addressed. Thereby, we pursue the approaches of both Maier and Mattheij and contribute novel aspects and solution approaches to the issue of adaptive multiple shooting.

6 Problems with Control Constraints

Goal of the previous chapter was not only to transfer the two main shooting approaches known from ODE optimal control to the parabolic PDE context and discuss the additional difficulties arising in this new framework, but also to enlighten the relationship between IMS and DMS in an abstract optimization environment. Up to now, only problems without any constraints besides the governing PDE were considered. In many cases, as is known from ODE optimal control, there are additional conditions to be fulfilled by the control or even the state variables. These extra requirements often render the problem at hand much more difficult. Special techniques have to be employed due to, e. g., lacking regularity properties. This chapter is dealing with control constrained problems and our intention is to compare IMS and DMS in the presence of such constraints. In Section 6.1, the formulation of the global OCP is extended to the case of control box constraints and afterwards the problem is decomposed and tailored to the multiple shooting framework. Section 6.2 is concerned with both IMS, where the corresponding results have been published in our article, Carraro et al. [22], and DMS in the control constrained context. The considered box constraints, i. e., functions which constitute an upper and/or lower bound for the control, are an important class of control constraints. They are particularly suitable for a multiple shooting formulation due to their localizability. Numerical tests similar to those from Chapter 5 are displayed in Section 6.3 for both IMS and DMS, which enables a final comparison between the indirect and direct shooting approaches in Section 6.4.

6.1 Problem reformulation

So far, OCP of the following form were considered:

$$\min_{(q,u)} J(q,u), \tag{6.1}$$

subject to the parabolic PDE constraint

$$\begin{aligned} \partial_t u(x,t) + \mathcal{A}(u)(x,t) + \mathcal{B}(q)(x,t) &= f(x,t) \quad \text{in } \Omega \times I, \\ u(x,0) &= u_0(x) \quad \text{in } \Omega, \end{aligned} \tag{6.2}$$

with suitable boundary conditions on $\partial\Omega \times I$. Now, we consider a more general case where additional constraints are imposed on the control variable $q(x,t)$. An important type of constraints occurring in many applications is given by so-called 'box constraints':

$$q_-(x,t) \leq q(x,t) \leq q_+(x,t). \tag{6.3}$$

Here, $q_-(x,t)$ and $q_+(x,t)$ are functions in $L^2(I;L^2(\Omega))$; if they are constant functions, the control is forced to remain between two constant bounds. This reminds of a rectangle or cuboid, where the notion of box constraints stems from. However, $q_-(x,t)$ and $q_+(x,t)$ may constitute more general functions. Figure 6.1 illustrates the idea of constant box constraints.

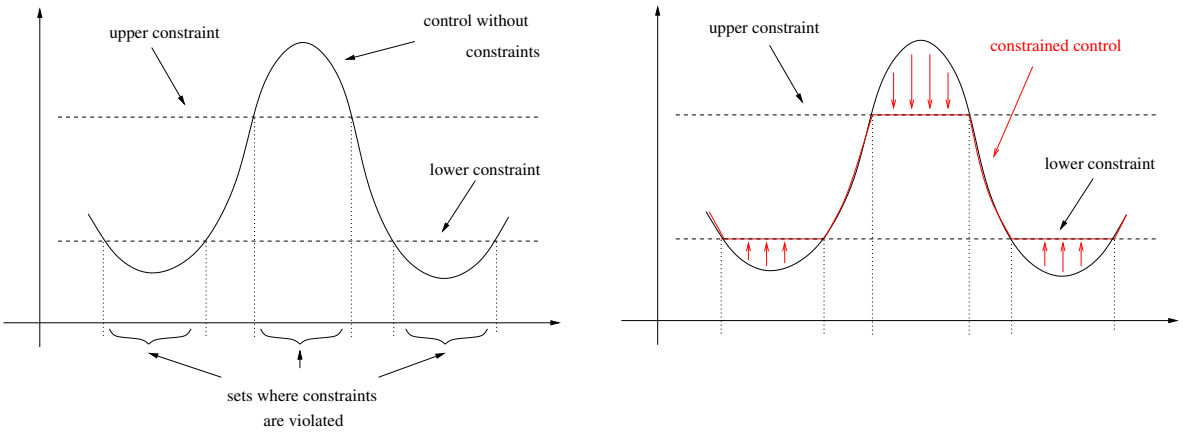

Figure 6.1. The unconstrained control variable and sets where the upper or lower constraint is violated (left); the constrained control variable after projection onto the admissible set (right).

Remark 6.1. From the literature, other types of control constraints such as

$$\int_{\Omega\times I} q(x,t)\,\mathrm{d}x\,\mathrm{d}t \le c,$$

(i. e., a constraint on the average of the control) are known. We do not consider such constraints in this work. As such global constraints cannot be easily localized to the shooting intervals, their treatment in the multiple shooting framework is more difficult than that of box constraints.

Imposing the additional control constraints (6.3) leads to a restricted set of feasible control functions. These potential control solutions are called 'admissible' for the problem and the set of admissible control functions is given by

$$Q_{\mathrm{ad}} = \{q \in Q \,|\, q_-(x,t) \le q(x,t) \le q_+(x,t)\}, \tag{6.4}$$

where the inequalities have to hold for almost all $(x,t) \in \Omega \times I$ and $q_-, q_+ \in Q$ are given functions satisfying $q_- < q_+$. As there holds

$$q_- = \lambda q_- + (1-\lambda)q_- < \lambda q_- + (1-\lambda)q_+ < \lambda q_+ + (1-\lambda)q_+ = q_+ \quad \text{for } \lambda \in (0,1),$$

every convex combination of q_- and q_+ is in Q_{ad} and thus, the admissible set is a convex subset of Q. In compact form, our constrained OCP thus reads

$$\min_{q\in Q_{\mathrm{ad}}, u\in X} J(q,u) \quad \text{subject to the weak formulation of (6.2).} \tag{6.5}$$

Results on existence and uniqueness of problems such as (6.5) can be found, e. g., in the textbooks of Hinze et al. [59] or Tröltzsch [108]. The discussion of these theoretical issues is kept shorter than in the unconstrained case which was covered in Chapter 3, as part of the theory in Section 3.2 already covers the case of constrained controls. The unique solvability of the side condition (6.2) is always assumed, which enables to define a solution operator $S : Q_{\mathrm{ad}} \to X, S(q) = u$. This in turn permits the definition of a reduced cost functional

$$\hat{J}(q) := J(q, S(q)), \tag{6.6}$$

which allows for the formulation of an unconstrained control problem on Q_{ad}:

$$\min_{q \in Q_{\mathrm{ad}}} \hat{J}(q). \tag{6.7}$$

This reduced problem enables to transfer the results from Chapters 3 and 5 to the control constrained case. It is revisited in the formulation of concrete shooting algorithms in Section 6.2.
In the presence of constraints of the form (6.3), the control equation $\mathcal{L}'_q(q, u, z)(\delta q) = 0$, concretized by (5.2c), has to be replaced by the variational inequality

$$\mathcal{L}'_q(q, u, z)(\delta q - q) \geq 0 \quad \forall \delta q \in Q_{\mathrm{ad}}. \tag{6.8}$$

The convexity of the set Q_{ad} is crucial. As a minimizer can lie on the boundary of Q_{ad}, the variational inequality (6.8) postulates the nonnegativity of all directional derivatives pointing into the set of admissible controls. If the minimizer lies in the interior of Q_{ad}, the variational inequality coincides with the original control equation, as then all directional derivatives and hence the complete gradient have to vanish. The state and adjoint equations remain the same as in system (5.2), thus the KKT system of (6.1) – (6.3) is given by

$$\begin{aligned}
((\partial_t u, \delta z)) + a(u)(\delta z) + b(q)(\delta z) - ((f, \delta z)) + (u(0) - u_0, \delta z(0)) &= 0, && \text{(6.9a)}\\
J'_u(q, u)(\delta u) - ((\partial_t z, \delta u)) + a'_u(u)(\delta u, z) + (z(T), \delta u(T)) &= 0, && \text{(6.9b)}\\
J'_q(q, u)(\delta q - q) + b'_q(q)(\delta q - q, z) &\geq 0. && \text{(6.9c)}
\end{aligned}$$

The treatment of the control inequality is not straightforward and makes the optimality system (6.9) more complicated. Methods that deal with variational inequalities are introduced in Section 6.2, but first the problem (6.1) – (6.3) is reformulated in terms that enable the application of a multiple shooting technique. In several steps, the control inequality (6.8) is transferred into a set of equations, which prepares the active set strategies presented in Section 6.2.
We define the sets $\mathcal{A}_-$ and $\mathcal{A}_+$ as

$$\begin{aligned}
\mathcal{A}_- &:= \{(x, t) \in \Omega \times I \mid q(x, t) = q_-(x, t)\},\\
\mathcal{A}_+ &:= \{(x, t) \in \Omega \times I \mid q(x, t) = q_+(x, t)\}.
\end{aligned} \tag{6.10}$$

The optimal control q coincides with either the lower or the upper constraint function on these sets. The respective constraint is then said to be 'active'. The subset of the domain $\Omega \times I$ where neither constraint is active is called the inactive set,

$$\mathcal{I} := (\Omega \times I) \setminus \{\mathcal{A}_- \cup \mathcal{A}_+\}. \tag{6.11}$$

As Q is a Hilbert space and $\mathcal{L}'_q(q,u,z)(\cdot) : Q \to \mathbb{R}$ is a linear functional, the Riesz representation theorem enables the following identification:

$$(\!(\mu, \delta q)\!) := -b'_q(q)(\delta q, z) - J'_q(q,u)(\delta q) = -\mathcal{L}'_q(q,u,z)(\delta q) \quad \forall \delta q \in Q. \tag{6.12}$$

This yields an additional Lagrange multiplier $\mu(x,t) \in Q$. The following conditions constitute an equivalent reformulation of inequality (6.8) which can be proven by means of a case-by-case analysis:

$$\begin{aligned} \mu(x,t) < 0 \quad &\text{for a. e.} \quad (x,t) \in \mathcal{A}_-, \\ \mu(x,t) > 0 \quad &\text{for a. e.} \quad (x,t) \in \mathcal{A}_+, \\ \mu(x,t) = 0 \quad &\text{for a. e.} \quad (x,t) \in \mathcal{I}. \end{aligned} \tag{6.13}$$

This means that if q is a minimizer and thus (6.8) is fulfilled, then μ is negative where the lower constraint is active, μ is positive where the upper constraint is active, and μ vanishes on the inactive set. After defining the negative and positive parts of μ,

$$\mu_-(x,t) := -\min\{\mu(x,t), 0\}, \quad \mu_+(x,t) := \max\{\mu(x,t), 0\}, \tag{6.14}$$

the Lagrange functional $\mathcal{L}(\cdot,\cdot,\cdot)$ can be complemented by two additional terms, the so-called complementarity conditions. The resulting expression constitutes a suitable extension of the Lagrangian for the control constrained OCP and is given by

$$\begin{aligned} \mathcal{L}(q,u,z,\mu_-,\mu_+) := J(q,u) + (\!(\partial_t u, z)\!) + a(u)(z) + b(q)(z) \\ + (u(0) - u_0, z(0)) - (\!(f,z)\!) + (\!(\mu_-, q_- - q)\!) + (\!(\mu_+, q - q_+)\!). \end{aligned} \tag{6.15}$$

The $\max\{\cdot\}$ and $\min\{\cdot\}$ functions are not differentiable in the classical sense. Therefore, deriving the KKT system by differentiating the extended Lagrangian (6.15) requires additional preparation. A suitable framework is given by the notion of Newton differentiability. We explain this concept in detail in the framework of primal-dual active set strategies in Section 6.2. In this generalized differentiability context, the Lagrange functional (6.15) can be differentiated with respect to all its arguments, resulting in the following first-order necessary optimality conditions for the control constrained case:

$$(\!(\partial_t u, \delta z)\!) + a(u)(\delta z) + b(q)(\delta z) - (\!(f, \delta z)\!) + (u(0) - u_0, \delta z(0)) = 0, \tag{6.16a}$$

$$J'_u(q,u)(\delta u) - (\!(\partial_t z, \delta u)\!) + a'_u(u)(\delta u, z) - (z(T), \delta u(T)) = 0, \tag{6.16b}$$

$$J'_q(q,u)(\delta q) + b'_q(q)(\delta q, z) - (\!(\mu_-, \delta q)\!) + (\!(\mu_+, \delta q)\!) = 0, \tag{6.16c}$$

$$(\!(q_- - q, \delta \mu_-)\!) = 0, \tag{6.16d}$$

$$(\!(q - q_+, \delta \mu_+)\!) = 0, \tag{6.16e}$$

$$\mu_-, \mu_+ \geq 0. \tag{6.16f}$$

These equations have to hold for all variations $\delta u, \delta z \in X$, $\delta q \in Q$, $\delta \mu_- \in Q_-$ and $\delta \mu_+ \in Q_+$, where Q_- and Q_+ are the following two subsets of the control space :

$$\begin{aligned} Q_- &:= \{q \in Q \mid q = 0 \text{ for a. e. } (x,t) \in \Omega \setminus \mathcal{A}_-\}, \\ Q_+ &:= \{q \in Q \mid q = 0 \text{ for a. e. } (x,t) \in \Omega \setminus \mathcal{A}_+\}. \end{aligned}$$

The last two inequalities (6.16f) must be fulfilled in almost every point $(x,t) \in \Omega \times I$. The system (6.16) has the structure of a BVP with separated boundary conditions, equivalently to the unconstrained case (cf. equation (3.32)). Therefore, it can be treated by means of multiple shooting, which is now discussed in a function space setting.
Building a suitable framework for multiple shooting methods for problems with control box constraints is similar to the proceeding in Section 3.4. Starting from the full Lagrangian (6.15), the same steps have to be taken as in the unconstrained case. The constraint functions $q_-, q_+ \in Q$ have to be split up to the subintervals I_j:

$$Q_j \ni q_-^j := q_-|_{I_j}, \quad Q_j \ni q_+^j := q_+|_{I_j}.$$

Similarly, this splitting is repeated for the Lagrange multiplier $\mu \in Q$ defined in (6.12) as well as for its positive and negative parts given by (6.14). The introduction of corresponding intervalwise variables μ^j, μ_-^j, and μ_+^j serves to rewrite the occurring intervalwise variational inequalities

$$\mathcal{L}'_{q^j}(q^j, u^j, z^j)(\delta q - q^j) \geq 0 \quad \forall \delta q \in Q_{\mathrm{ad}}^j$$

as equivalent sets of equations. With these preparatory definitions, and using the functions s^j and λ^j as state and adjoint shooting variables as before in Section 3.4, the extension of the full Lagrangian, denoted by $\overline{\mathcal{L}}$, can be formulated as follows:

$$\begin{aligned}
\overline{\mathcal{L}}((q^j, u^j, z^j, \mu_-^j, \mu_+^j)_{j=0}^{M-1}, (s^j, \lambda^j)_{j=0}^{M}) &:= \kappa_1 \sum_{j=0}^{M-1} J_1(u^j) + \kappa_2 J_2(u^{M-1}(\tau_M)) \\
&+ \frac{\alpha}{2} \sum_{j=0}^{M-1} \int_{I_j} \|q^j\|^2 \, dt + \sum_{j=0}^{M-1} \Big[(\!(\partial_t u^j, z^j)\!) + a(u^j)(z^j) + b(q^j)(z^j) - (\!(f|_{I_j}, z^j)\!) \Big] \\
&+ \sum_{j=0}^{M-1} (u^j(\tau_j) - s^j, z^j(\tau_j)) + \sum_{j=0}^{M-1} (s^{j+1} - u^j(\tau_{j+1}), \lambda^{j+1}) + (s^0 - u_0, \lambda^0) \\
&+ \sum_{j=0}^{M-1} \Big[(\!(\mu_-^j, q_-^j - q^j)\!) + (\!(\mu_+^j, q^j - q_+^j)\!) \Big].
\end{aligned} \tag{6.17}$$

This is the Lagrangian associated with the OCP given by (3.43) – (3.44) and the additional intervalwise box constraints

$$q_-^j(x,t) \leq q^j(x,t) \leq q_+^j(x,t). \tag{6.18}$$

Differentiation of $\overline{\mathcal{L}}$ with respect to z^j, u^j, and q^j in the directions $(\delta z, \delta u, \delta q) \in X_j \times X_j \times Q_j$ leads to the intervalwise BVP (3.48a) – (3.48c), where the control equation has to be slightly modified as follows:

$$\alpha(\!(q^j, \delta q)\!) + b'_{q^j}(q^j)(\delta q, z^j) - (\!(\mu_-^j, \delta q)\!) + (\!(\mu_+^j, \delta q)\!) = 0. \tag{6.19}$$

Furthermore, differentiating $\overline{\mathcal{L}}$ with respect to the shooting variables s^j and λ^j results in the system of equations (3.48d) – (3.48g), which is exactly the same as in the unconstrained

case. The main difference compared to the unconstrained case consists in the equations resulting from the differentiation of $\overline{\mathcal{L}}$ with respect to μ_-^j and μ_+^j:

$$(\!(q_-^j - q^j, \delta\mu_-)\!) = 0, \quad (\!(q^j - q_+^j, \delta\mu_+)\!) = 0, \quad \mu_-^j, \mu_+^j \geq 0. \tag{6.20}$$

These conditions correspond to (6.16d) – (6.16f).
With this preparatory reformulation of the control constrained OCP in the multiple shooting framework, we are able to state two pseudo-algorithms for IMS and DMS. They are the respective starting points for Subsections 6.2.1 and 6.2.2 (see Algorithms 6.1 and 6.4).

6.2 Multiple shooting for control constrained problems

There are different methods for treating control constrained PDE control problems; classical approaches are often based on projections of the unconstrained controls onto the admissible set Q_{ad}, whereas modern strategies rely on the concepts of active sets or Newton differentiability. This section introduces both classes of methods, but embeds them at once into the multiple shooting framework. Subsection 6.2.1 is concerned with IMS, where we provide a detailed presentation of the different approaches. In Subsection 6.2.2 the DMS case is covered, and the presentation stresses the differences to the IMS case and is otherwise kept short.

6.2.1 IMS for problems with control box constraints

This subsection describes the treatment of control constraints such as (6.3), extending different methods to the IMS framework which requires a splitting of the constraints as in (6.18). A brief survey of methods for control-constrained optimal control problems (including the ones presented here) can be found in Herzog & Kunisch [51].

Algorithm 6.1 IMS for optimal control problems with control constraints

Require: Decomposition $\overline{I} = \{0\} \cup \bigcup_{j=0}^{M-1} I_j$, initial values $\{(s_0^j, \lambda_0^{j+1})_{j=0}^{M-1}\}$.
1: Set $k = 1$.
2: **while** Shooting conditions (3.48d) – (3.48g) not fulfilled **do**
3: **for** $j = 0$ to $M - 1$ **do**
4: Solve intervalwise BVP (3.48a) – (3.48c).
5: **if** Control constraints imposed **then**
6: Account for (6.19) and conditions (6.20), i. e., compute constrained controls $\tilde{q}_k^j$.
7: **end if**
8: **end for**
9: Solve (3.48d) – (3.48g), compute initial value update $\{(s_k^j, \lambda_k^{j+1})_{j=0}^{M-1}\}$, set $k \leftarrow k+1$.
10: **end while**

The control constrained OCP has been adapted to the requirements of multiple shooting, enabling us to start from Algorithm 6.1 which is an extension of Algorithm 5.1. Then, both projection methods and the modern primal-dual active set strategy are discussed.

Gradient projection method. If intervalwise box constraints of type (6.18) are imposed on the subinterval OCPs, Algorithm 5.2 may produce iterates q_k^j of the control that violate the constraints, i. e., $q_k^j \notin Q_{\text{ad}}^j$. In this case, it must be ensured that the algorithm corrects this deficiency. This is done most easily by projecting non-admissible iterates onto the set Q_{ad}^j using the projection operator defined by

$$P_{Q_{\text{ad}}^j}(q_k^j) = P_{[q_{k,-}^j, q_{k,+}^j]}(q_k^j) = \max\{q_{k,-}^j, \min\{q_k^j, q_{k,+}^j\}\}. \tag{6.21}$$

Algorithm 5.2 is then extended by the projected gradient algorithm in its following form:

Algorithm 6.2 Projection of nonadmissible control iterates onto Q_{ad}^j

Require: Set $k = 0$, initial control $q_0^j \in Q_{\text{ad}}^j$.
1: Perform algorithm 5.2.
2: **if** q_{k+1}^j in step 10 of algorithm 5.2 is $\notin Q_{\text{ad}}^j$ **then**
3: Determine a step length σ such that $\hat{J}(P_{Q_{\text{ad}}^j}(q_{k+1}^j - \sigma \nabla \hat{J}(q_{k+1}^j))) < \hat{J}(q_{k+1}^j)$.
4: Set $\tilde{q}_{k+1}^j = P_{Q_{\text{ad}}^j}(q_{k+1}^j - \sigma \nabla \hat{J}(q_{k+1}^j))$.
5: **end if**

Steps 4–7 of Algorithm 6.1 can now be replaced by Algorithm 6.2 for control constrained problems. Usually the determination of the step size in step 3 is carried out by applying a projected Armijo rule (for details, see Hinze et al. [59]). The projected gradient method is globally convergent but only with a linear rate which was shown, e. g., by Dunn [36].

Remark 6.2. The definition of $\tilde{q}_{k+1}^j$ as in step 4 of Algorithm 6.2 could be replaced by the more general

$$\tilde{q}_{k+1}^j = P_{Q_{ad}^j}(q_{k+1}^j - \sigma H^{-1} \nabla j(q_{k+1}^j)). \tag{6.22}$$

This reminds of a Newton-type method, where H is the reduced Hessian $\nabla^2 j(q_{k+1}^j)$ or an approximation of it. Note that, if H is chosen as the identity, the original variant from Algorithm 6.2 is restored. Methods based on the concept (6.22), using the Hessian with modifications in those blocks corresponding to degrees of freedom where the constraints are active, are known as 'projected Newton methods' and are applied to solve control problems in practice (see, e. g., Kelley & Sachs [66]). Although they often display a superlinear convergence behavior and therefore outperform the above projected gradient method, it is known that they are not generally convergent. A counterexample is given by Kelley [65]. Therefore, they cannot always be applied.

The implementation of the control constrained examples presented in Section 6.3 below is based on projection methods. Nevertheless, we elaborate the primal-dual active set strategy in the context of multiple shooting methods as well.

Primal-dual active set strategy. In the past fifteen years active set strategies involving both state and adjoint variables have been thoroughly examined as solution techniques for constrained OCP. They have been first described by Bergounioux et al. [9] for elliptic OCP and were applied, e. g., by Griesse & Vexler [46] and Vexler & Wollner [112]. In the parabolic case, a similar procedure has been suggested by Kunisch & Rösch [72] and was used, e. g., by Griesse & Vexler [46] and Griesse & Volkwein [47]. The primal-dual active set strategy, which is equivalent to a semi-smooth Newton method, constitutes an alternative to projected gradient or Newton methods. As Remark 6.2 indicates, one has to be careful when employing projection methods. We state below that semi-smooth Newton methods display a superlinear convergence behavior (cf. Hintermüller et al. [57]). Using the notation introduced in the previous section, we present an active set strategy in its semi-smooth Newton formulation.

The easiest way to derive a primal-dual active set algorithm starts from the system of conditions (6.13) on the additional Lagrange multiplier μ. As $\mu(x,t) < 0$ almost everywhere on $\mathcal{A}_-$, we infer that

$$\mu(x,t) + cq(x,t) = \mu(x,t) + cq_-(x,t) < cq_-(x,t) \tag{6.23}$$

for all constants $c > 0$ and almost all $(x,t) \in \mathcal{A}_-$. An analogous lower bound on $\mu + cq$ is valid on $\mathcal{A}_+$. Therefore, the active sets (6.10) can be expressed in the following alternative form:

$$\begin{aligned} \mathcal{A}_- &= \{(x,t) \in \Omega \times I \mid cq(x,t) + \mu(x,t) \leq cq_-(x,t)\}, \\ \mathcal{A}_+ &= \{(x,t) \in \Omega \times I \mid cq(x,t) + \mu(x,t) \geq cq_+(x,t)\}. \end{aligned} \tag{6.24}$$

First, we interpret equations (6.16c)–(6.16f) including the reduced gradient and the complementarity conditions in an intervalwise manner:

$$J_q'(q^j,u^j)(\delta q) + b_q'(q^j)(\delta q, z^j) - (\!(\mu_-^j, \delta q)\!) + (\!(\mu_+^j, \delta q)\!) = 0, \tag{6.25a}$$

$$(\!(q_-^j - q^j, \delta\mu_-)\!) = 0, \quad (\!(q^j - q_+^j, \delta\mu_+)\!) = 0, \quad \mu_-^j, \mu_+^j \geq 0. \tag{6.25b}$$

Here, superscripts j indicate restrictions of global functions to the subinterval I_j. The equalities in (6.25b) are equivalent to the complementarity conditions

$$(\!(q_-^j - q^j, \mu_-^j)\!) = 0, \quad (\!(q^j - q_+^j, \mu_+^j)\!) = 0. \tag{6.26}$$

To avoid dealing with inequalities, we use the concept of complementarity functions.

Definition 6.1. *A function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is called a complementarity function if the following condition holds:*

$$\varphi(a,b) = 0 \quad \Longleftrightarrow \quad a \leq 0,\ b \leq 0,\ a \cdot b = 0.$$

As $\varphi(a,b) = \max\{a,b\}$ is an example for a complementarity function, and with the equality $\max\{a,b\} = a + \max\{0, b-a\}$, the complementarity conditions (6.26) can be reformulated as a system of two equations,

$$J_q'(q^j,u^j)(\delta q) + b_q'(q^j)(\delta q, z^j) + (\!(\mu^j, \delta q)\!) = 0, \tag{6.27a}$$

$$\mu^j - \max\{0, \mu^j + c(q^j - q_+^j)\} - \min\{0, \mu^j + c(q^j - q_-^j)\} = 0, \tag{6.27b}$$

where we used the notation $\mu^j := \mu_+^j - \mu_-^j$ as well as transformations of the min / max functions. The system (6.27) includes the box constraints and is equivalent to (6.25) for all $c > 0$. Below in Algorithm 6.3, after the state and adjoint equations are solved and u^j and z^j are therefore known, we will abbreviate (6.27) by $G(q^j, \mu^j) = 0$. With these equations, the subinterval KKT system is completely reformulated as a system of equations, and the original inequalities are expressed by the min / max functions:

$$(\!(\partial_t u^j, \delta z)\!) + a(u^j)(\delta z) + b(q^j)(\delta z) - (\!(f|_{I_j}, \delta z)\!) + (u^j(\tau_j) - s^j, \delta z(\tau_j)) = 0, \tag{6.28a}$$

$$J_u'(q^j, u^j)(\delta u) - (\!(\partial_t z^j, \delta u)\!) + a_u'(u^j)(\delta u, z^j) - (z^j(\tau_{j+1}), \delta u(\tau_{j+1})) = 0, \tag{6.28b}$$

$$J_q'(q^j, u^j)(\delta q) + b_q'(q^j)(\delta q, z^j) - (\!(\mu_-^j, \delta q)\!) + (\!(\mu_+^j, \delta q)\!) = 0, \tag{6.28c}$$

$$\mu^j - \max\{0, \mu^j + c(q^j - q_+^j)\} - \min\{0, \mu^j + c(q^j - q_-^j)\} = 0. \tag{6.28d}$$

Application of Newton's method to solve the KKT conditions is tempting. However, the min / max functions are not differentiable in the classical sense. Therefore, we have to use the more general concept of slant (or Newton) differentiability which applies to min / max (see Hintermüller, Ito & Kunisch [57] or Ito & Kunisch [60]).

Definition 6.2. *Let X and Y be Banach spaces. A mapping $F : D \subset X \to Y$ is called Newton differentiable on a set $U \subset D$ if there is a family of mappings $F' : U \to \mathcal{L}(X, Y)$ such that for each $x \in U$ it holds*

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

The family of functions F' is called the Newton derivative of F.

Newton differentiability is no pointwise differentiabilty concept and the Newton derivative is generally not unique. This differentiability notion enables the definition of semismooth functions:

Definition 6.3. *Let X and Y be Banach spaces. A function $F : D \subset X \to Y$ on an open set D is called semismooth in a point x if F is Newton differentiable in x with Newton derivative F' and if the limit $\lim_{t \searrow 0} F'(x + th)h$ exists uniformly in h with $\|h\| = 1$.*

These notions enable the definition of a variant of Newton's method for semismooth functions. A root x^* of the semismooth function $F(x)$ can be approximated iteratively by choosing a starting point x_0 and applying the semismooth Newton formula

$$F'(x_n)x_{n+1} = F'(x_n)x_n - F(x_n).$$

This variant of Newton's method displays a superlinear convergence behavior. In the following, we derive the primal-dual active set strategy based on a semismooth Newton approach for a reduced formulation of the control constrained problem. Therefore, we define local active sets in analogy to (6.24) as well as local inactive sets on the shooting

intervals I_j:

$$\begin{aligned}\mathcal{A}_-^j &:= \{(x,t) \in \Omega \times I_j \mid \mu^j(x,t) + c(q^j(x,t) - q_-^j(x,t)) > 0\},\\ \mathcal{A}_+^j &:= \{(x,t) \in \Omega \times I_j \mid \mu^j(x,t) + c(q(x,t)^j - q_+^j(x,t)) < 0\},\\ \mathcal{A}^j &:= \mathcal{A}_-^j \cup \mathcal{A}_+^j, \quad \mathcal{I}^j := (\Omega \times I_j) \setminus \mathcal{A}^j.\end{aligned} \tag{6.29}$$

The sets $\mathcal{A}_+^j$ and $\mathcal{A}_-^j$ describe the subdomains of $\Omega \times I_j$ where the max and min of (6.27b) are attained, respectively. These sets are required because the Jacobian of (6.27) depends on them via their characteristic functions $\chi_{\mathcal{A}^j}, \chi_{\mathcal{I}^j}$, which can be seen from the following formulation of the Newton system at a given iterate $(q_k^j, \mu_k^j)^\top$:

$$\begin{bmatrix} \nabla^2 \hat{J}(q_k^j) & I \\ c\chi_{\mathcal{A}_k^j} & -\chi_{\mathcal{I}_k^j} \end{bmatrix} \begin{bmatrix} \delta q \\ \delta \mu \end{bmatrix} = - \begin{bmatrix} \nabla \hat{J}(q_k^j) + \mu_k^j \\ c\chi_{\mathcal{A}_{k,+}^j}(q_k^j - q_+^j) + c\chi_{\mathcal{A}_{k,-}^j}(q_k^j - q_-^j) - \chi_{\mathcal{I}_k^j}\mu_k^j \end{bmatrix}. \tag{6.30}$$

Recall that in terms of the reduced problem,

$$\hat{J}_q'(q^j)(\delta q) = J_q'(q^j, u^j)(\delta q) + b_q'(q^j)(\delta q, z^j)$$

due to (3.30), which is the connection to (6.27). As before, we want to solve (6.30) in a matrix-free way. To achieve this, we simply have to go through step 8 of Algorithm 5.2 for the left upper block of the Jacobian in (6.30), while the remaining blocks can be treated easily thanks to their simple structure. Now we have everything at hand to formulate the primal-dual active set strategy in its semismooth Newton variant for the reduced problem. Again, we can replace steps 4–7 of Algorithm 6.1 by the following Algorithm 6.3.

Algorithm 6.3 Active set resp. semi-smooth Newton algorithm for reduced problem

Require: Set $k = 0$, prescribe tolerance TOL_1 and initial values q_0^j, μ_0^j.
1: **while** $\|G(q_k^j, \mu_k^j)\| > TOL_1$ **do**
2: Solve state equation (3.48a).
3: Solve adjoint equation (3.48b).
4: Compute $G(q_k^j, \mu_k^j)$.
5: Determine active sets (6.29).
6: Set $i = 0$, prescribe tolerance TOL_2 and $\delta q_{k,0}^j$.
7: **while** $\|(\delta q_{k,i+1}^j, \delta \mu_{k,i+1}^j)^\top - (\delta q_{k,i}^j, \delta \mu_{k,i}^j)^\top\| > TOL_2$ **do**
8: Solve system (6.30) by a matrix-free method.
9: **end while**
10: Set $k \leftarrow k+1$ and compute updates $q_{k+1}^j = q_k^j + \delta q_{k,end}^j$, $\mu_{k+1}^j = \mu_k^j + \delta \mu_{k,end}^j$.
11: **end while**

Remark 6.3. In step 1 of Algorithm 6.3, we choose a stopping criterion similar to the corresponding one in Algorithm 6.1. Instead, one could terminate the algorithm when the active sets in two subsequent iterations coincide, i. e., when $\mathcal{A}_{k,+}^j \equiv \mathcal{A}_{k+1,+}^j$ and $\mathcal{A}_{k,-}^j \equiv \mathcal{A}_{k+1,-}^j$. This latter criterion is commonly used in the description of active set methods.

6.2.2 DMS for problems with control box constraints

In contrast to the IMS approach, in the DMS case the control is part of the shooting system and therefore the control constraints are accounted for only after this system is solved. Analogously to Algorithm 6.1 in the previous subsection, we provide a draft of the proceeding in Algorithm 6.4. Then we adapt the projection method introduced in the IMS context in order to concretize steps 7–9 of this algorithm.

Algorithm 6.4 DMS for optimal control problems with control constraints

Require: Decomposition $\overline{I} = \{0\} \cup \bigcup_{j=0}^{M-1} I_j$, initial values $\{(s_0^j, q_0^j, \lambda_0^{j+1})_{j=0}^{M-1}\}$.
1: Set $k = 1$.
2: **while** Shooting conditions (3.48c) – (3.48g) not fulfilled **do**
3: **for** $j = 0$ to $M - 1$ **do**
4: Solve intervalwise IVP (3.48a) – (3.48b).
5: **end for**
6: Solve (3.48c) – (3.48g), compute initial value update $\{(s_k^j, q_k^j, \lambda_k^{j+1})_{j=0}^{M-1}\}$, set $k \leftarrow k + 1$.
7: **if** Control constraints imposed **then**
8: Account for (6.19) and conditions (6.20), i. e., compute constrained controls $\tilde{q}_k^j$.
9: **end if**
10: **end while**

The application of a projected gradient method within the original DMS Algorithm 5.4 is straightforward and displayed in the following Algorithm 6.5. Note that the control function $\mathbf{q}_k$ comprises all subinterval controls q_k^j and can be interpreted as a global control function. However, Algorithm 6.5 could be formulated equivalently for the controls on the different shooting intervals.

Algorithm 6.5 Projection of nonadmissible control updates onto Q_{ad}

1: Perform Algorithm 5.4, replacing step 6 by Algorithm 5.5 for solving the shooting system.
2: **if** $\mathbf{q}_{k+1}$ in step 7 of Algorithm 5.5 is $\notin Q_{\text{ad}}$ **then**
3: Determine a step length σ such that $\hat{J}(P_{Q_{\text{ad}}}(\mathbf{q}_{k+1} - \sigma \nabla \hat{J}(\mathbf{q}_{k+1}))) < \hat{J}(\mathbf{q}_{k+1})$.
4: Set $\tilde{\mathbf{q}}_{k+1} = P_{Q_{\text{ad}}}(\mathbf{q}_{k+1} - \sigma \nabla \hat{J}(\mathbf{q}_{k+1}))$.
5: **end if**

Remark 6.4. In the DMS framework, the projection of the intervalwise controls q_k^j onto the set Q_{ad} of admissible controls is carried out after the shooting system has been solved. It is therefore possible to interpret the combination of all subinterval control functions as one global function $\mathbf{q}_k$ on the complete solution interval I. In principle, DMS does not require the localizability of the control constraints which is crucial for the IMS method. As the control constraints can be imposed globally, DMS is presumably better suited to handle constraints of a more global type such as the mean value constraints introduced in Remark 6.1. Although this issue is not further elaborated in this thesis, we suppose that techniques for global constraints can be applied in the DMS context.

6.3 Numerical tests

This section serves to verify the theoretical results achieved in Sections 6.1 and 6.2 by means of a numerical example. Therefore, some of the test problems already discussed in Section 5.5 are complemented by additional control box constraints. We do not reconsider linear problems in this context, but display only results for nonlinear examples. The tests enable an extended comparison of IMS and DMS presented in the next section. Furthermore, the results again motivate the development of shooting grid adaptation techniques in Chapter 7. In programming the code for the examples, projection methods are realized; although they are less efficient than the modern active set strategies, they are easier to implement. The computations are based on the discretization concepts from Section 4.1; again, the finite element software `deal.ii` was used to obtain them. In order to provide meaningful results on computing times, we emphasize again that all computations were carried out on the same computer.

6.3.1 Results for IMS

We reconsider Example 5.2 which is now complemented with control box constraints. The problem consists of a distributed tracking-type functional subject to a nonlinear PDE. The latter depends on a parameter ω and it was shown in Section 5.5 for a corresponding linear problem that for certain choices of this parameter, simple shooting becomes unstable.

Example 6.1. *We consider the problem*

$$\min_{(q,u)} \; J(q,u) = \frac{1}{2}\int_0^T \|u(x,t) - \hat{u}(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t + \frac{\alpha}{2}\int_0^T \|q(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t,$$

subject to the nonstationary nonlinear Helmholtz problem

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + u^3(x,t) &= q(x,t) && \text{in } \Omega \times (0,T], \\ u(x,t) &= 0 && \text{on } \partial\Omega \times [0,T], \\ u(x,0) &= u_0(x) && \text{in } \Omega \end{aligned}$$

and the constant box constraints

$$-0.5 \le q(x,t) \le 0.5 \;\; \text{a. e. in } \Omega \times [0,T].$$

The computational domain $\Omega = (-1,1)^2$ and the end time $T = 5$ are as before. The regularization parameter is fixed at $\alpha = 0.5$, and the Helmholtz parameter is varied, $\omega \in \{3,4,5,6,7\}$. Furthermore, the tracking function is again given by

$$\hat{u}(x,t) := \begin{cases} \frac{2}{5}t \cdot \left(1 - x_1^{12}\right)\left(1 - x_2^{12}\right), & t \le \frac{5}{2}, \\ \left(\frac{2}{5}t - 2\right) \cdot \left(1 - x_1^{12}\right)\left(1 - x_2^{12}\right), & t > \frac{5}{2}. \end{cases}$$

The control box constraints lead to the set Q_{ad} of admissible controls,

$$Q_{\text{ad}} = \Big\{ q(x,t) \in Q = L^2(I; L^2(\Omega)) \mid -0.5 \le q(x,t) \le 0.5 \Big\}. \tag{6.31}$$

For simplicity, we choose constant box constraints, but without difficulty it is possible to replace the constant bounds in (6.31) by general L^2-functions $q_-(x,t)$ and $q_+(x,t)$, respectively. In Table 6.1, we illustrate the shooting iterates of IMS for Example 6.1.

Table 6.1. Example 6.1: Development of IMS solution on 10 equidistant shooting intervals without (left) and with global refinement (right) for $\omega = 7$ (required: $\|F\| < 1.0 \cdot 10^{-3}$). The computing times are 14737 seconds without and 7800 seconds with global refinement, corresponding to a time saving of 47.1%.

	without refinement			with refinement		
Newton step	$\#_{\text{GMRES}}$	$J(q,u)$	$\|F\|$	$\#_{\text{GMRES}}$	$J(q,u)$	$\|F\|$
1	–	2.190	$4.3 \cdot 10^{00}$	–	1.436	$1.6 \cdot 10^{00}$
2	26	1.811	$5.9 \cdot 10^{-01}$	19	1.338	$8.0 \cdot 10^{-01}$
3	25	2.270	$1.9 \cdot 10^{-01}$	25	2.239	$3.3 \cdot 10^{-01}$
4	25	2.158	$6.9 \cdot 10^{-02}$	25	2.193	$6.7 \cdot 10^{-02}$
5	25	2.197	$6.4 \cdot 10^{-03}$	25	2.200	$9.0 \cdot 10^{-03}$
6	25	2.193	$2.6 \cdot 10^{-03}$	25	2.194	$2.0 \cdot 10^{-03}$
7	25	2.195	$4.2 \cdot 10^{-04}$	25	2.195	$3.0 \cdot 10^{-04}$

The difference in the number of GMRES iterations per Newton step is not large. This holds irrespectively of whether the solution process is carried out completely on the finest spatial mesh or if the global refinement strategy proposed in Subsection 5.5.3 is included.

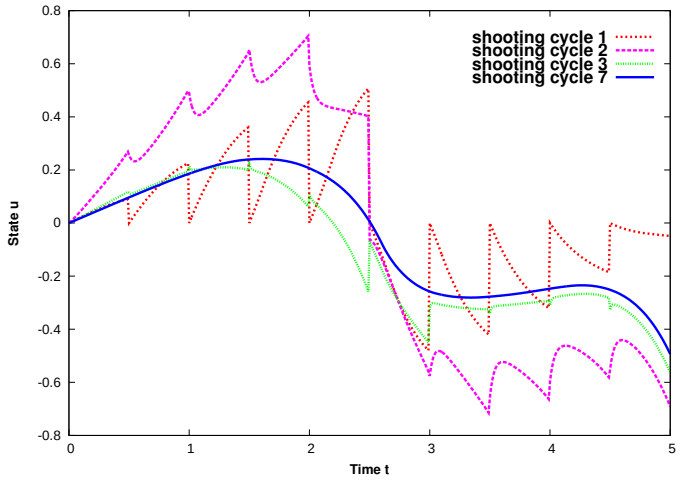

Figure 6.2. Example 6.1: The state variable $u(0,0,t)$ at different IMS cycles in the control constrained case.

Both approaches provide the same functional value after the shooting residual is sufficiently reduced. However, the refinement strategy saves almost half the computing time. Figure

6.2 displays the state variable at different multiple shooting iterations. It is noteworthy that in the control constrained case, the number of Newton iterations until convergence of the shooting process is almost doubled in comparison to the unconstrained case (see Figure 5.4). We remark that the computations were carried out without preconditioning, as the SGS preconditioner proposed in Subsection 5.1.2 turned out to be even less efficient than in the unconstrained case.

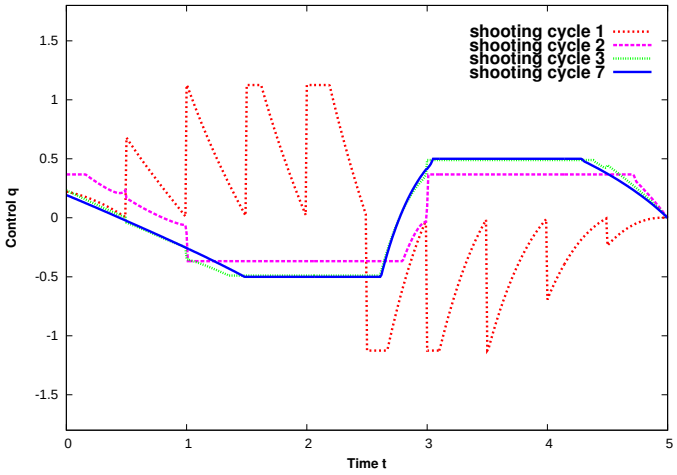

Figure 6.3. Example 6.1: The control $q(0, 0, t)$ at different IMS cycles in the control constrained case.

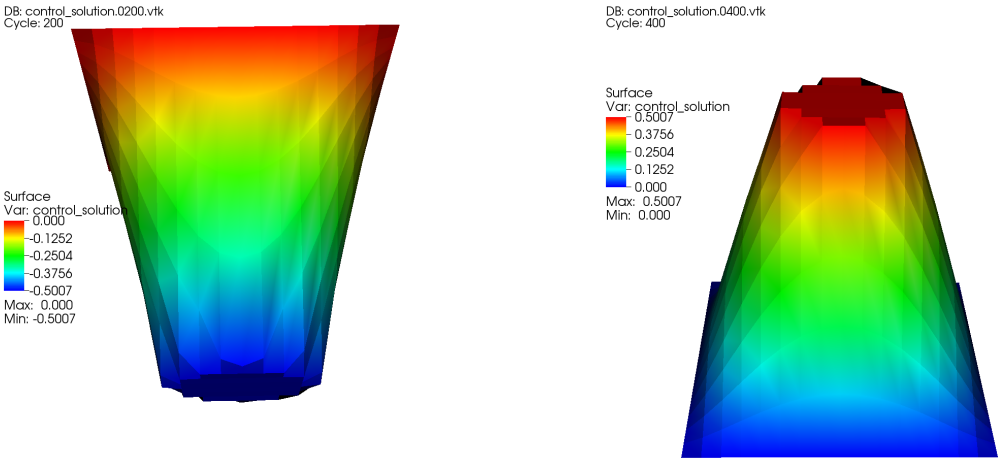

Figure 6.4. Example 6.1: The control variable at different timepoints: at $t = 2$, the lower constraint q_- is active; at $t = 4$, the upper constraint q_+ is active (see also Figure 6.3).

The control variable at different shooting cycles is depicted in Figure 6.3. The control constraints (6.31) are fulfilled except for the first shooting iteration. The first iteration is carried out on a coarse mesh of 4 cells, meaning there is only one spatial degree of freedom at $(0, 0, t)$. Therefore, the control cannot be appropriately projected onto the constraints. Apart from this difference in the control variable, a comparison of Figures 5.4 and 6.2, respectively Figures 5.5 and 6.3, provides only minor deviations in the state variable. In Figure 6.4, the constrained control at two concrete timepoints is displayed; for $t = 2$, which corresponds to timestep no. 200, the lower constraint $q_- = -0.5$ is active, whereas for $t = 4$, corresponding to timestep no. 400, the upper constraint $q_+ = 0.5$ is active.
The following results concern two aspects of multiple shooting that were already examined in Section 5.5 in the unconstrained case. Table 6.2 states the minimum number of equidistant shooting intervals required to solve the problem from Example 6.1 for each value of the Helmholtz parameter ω.

Table 6.2. Example 6.1: The minimum number of shooting intervals (SI) required for different values of the parameter ω. The number of timesteps per SI is chosen so that the total number of timesteps is as close as possible to 500 (required: $\|F\| < 1.0 \cdot 10^{-3}$).

ω	#SI	#ts/SI	#Newton	#GMRES	$J(q,u)$	$\|F\|$	t(s)
3	1	500	4	2–8	2.210	$5.7{\cdot}10^{-07}$	1275
4	3	167	4	5–10	1.997	$9.7{\cdot}10^{-04}$	4322
5	6	84	7	11–16	1.827	$4.8{\cdot}10^{-04}$	9063
6	8	63	6	15–21	1.943	$8.0{\cdot}10^{-04}$	6975
7	10	50	6	19–25	2.915	$3.0{\cdot}10^{-04}$	7800

While the problem is solvable by means of indirect simple shooting for $\omega = 3$, the value $\omega = 7$ requires 10 equidistant shooting intervals (cf. the results of Table 6.1). While the other computations were carried out with 500 timesteps equally distributed to the shooting intervals, some results of Table 6.2 require a deviation from this total number of timesteps. However, this deviation is kept as small as possible.

Finally, Table 6.3 confirms the results of Subsection 5.5.4, where we stated that in PDE governed OCP it is best to choose as few shooting intervals as possible, at least in terms of computing time. As this minimum number of shooting intervals is not known in advance, this raises the question of how to find a suitable number and distribution of shooting points adaptively in order to avoid time-consuming trial and error computations. We address this problem in Chapter 7.

6.3.2 Results for DMS

This subsection is structured similarly to the previous one on IMS in order to facilitate comparisons between the two shooting approaches. As the global refinement strategy proposed in Subsection 5.5.3 accelerates all previous computations, we employ it throughout this subsection. Furthermore, the following results have been achieved by using a damped

Table 6.3. Example 6.1: IMS with and without global refinement strategy for $\omega = 3$. Comparison of different numbers of shooting intervals (required: $\|F\| < 1.0 \cdot 10^{-3}$). The last column gives the saving in terms of computing time as compared to IMS without refinement strategy.

	without refinement				with refinement			
#SI	$\#_{\text{Newton(it)}}$	$J(q,u)$	$\|F\|$	t(s)	$\#_{\text{Newton(it)}}$	$J(q,u)$	$\|F\|$	%
1	2 (9)	2.210	$1.1 \cdot 10^{-7}$	1817	4 (2–8)	2.210	$5.7 \cdot 10^{-7}$	29.8
2	3 (9)	2.208	$8.2 \cdot 10^{-5}$	2710	4 (3–8)	2.207	$8.5 \cdot 10^{-4}$	53.2
4	3 (12)	2.208	$1.1 \cdot 10^{-4}$	2712	4 (7–11)	2.207	$9.0 \cdot 10^{-4}$	52.3
5	3 (13)	2.208	$2.2 \cdot 10^{-4}$	2712	4 (9–13)	2.208	$7.7 \cdot 10^{-4}$	51.2
10	3 (21)	2.208	$3.5 \cdot 10^{-4}$	2868	5 (17–21)	2.208	$1.2 \cdot 10^{-4}$	20.4
20	3 (48)	2.213	$5.1 \cdot 10^{-4}$	4493	5 (41–48)	2.208	$2.3 \cdot 10^{-4}$	21.9

Newton-GMRES method for the shooting system. The Newton update strategy is given as follows:

$$\begin{aligned} \mathbf{s}_{k+1} &= \mathbf{s}_k + \nu \cdot \delta \mathbf{s}_k^{end}, \\ \mathbf{q}_{k+1} &= \mathbf{q}_k + \nu \cdot \delta \mathbf{q}_k^{end}, \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \nu \cdot \delta \boldsymbol{\lambda}_k^{end}. \end{aligned} \tag{6.32}$$

The damping parameter is chosen as $\nu = 0.5$ as long as the residual norm is $\|F\| > 1$ and set to $\nu = 1$ (i. e., undamped Newton) after $\|F\|$ falls below this threshold.
We revisit Example 6.1 in the DMS context. Table 6.4 provides analogous results as Table 6.1 in the IMS framework. While requiring more shooting cycles as well as twice as

Table 6.4. Example 6.1: DMS for $\omega = 7$ on 10 shooting intervals discretized by 50 timesteps each. The solution involves global space mesh refinement and a damped Newton strategy for the shooting system (required: $\|F\| \leq 10^{-3}$). The total computing time is 4953 seconds.

ref. level	#Newton	#GMRES	$J(q,u)$	$\|F\|$
1	1	–	1.765	$6.1{\cdot}10^{00}$
1	2	28	1.503	$8.1{\cdot}10^{00}$
2	3	49	1.827	$4.7{\cdot}10^{00}$
3	4	49	1.981	$3.0{\cdot}10^{00}$
4	5	50	2.064	$1.8{\cdot}10^{00}$
4	6	50	2.115	$1.1{\cdot}10^{00}$
4	7	50	2.147	$6.3{\cdot}10^{-01}$
4	8	50	2.188	$1.0{\cdot}10^{-01}$
4	9	50	2.195	$6.9{\cdot}10^{-03}$
4	10	50	2.195	$4.4{\cdot}10^{-04}$

many GMRES iterations per shooting cycle than IMS, the solution process is nevertheless

significantly faster (4953 seconds as compared to 7800 seconds, which corresponds to a saving of 36.5%). Supposedly, this is due to the necessity in the IMS framework to solve a smaller variant of the original OCP on each shooting interval, which is computationally expensive. The corresponding solution of state and adjoint IVP in the DMS approach is comparably cheap.

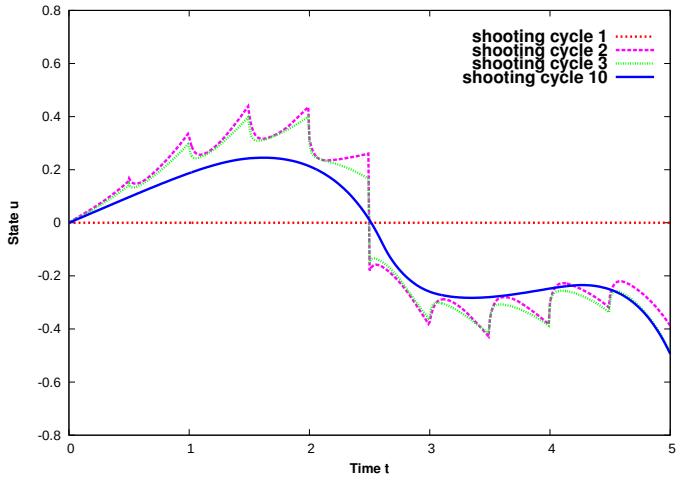

Figure 6.5. Example 6.1: The state variable $u(0,0,t)$ at different DMS cycles in the control constrained case for $\omega = 7$ and $\alpha = 0.5$.

Figures 6.5 and 6.6 display the development of state $u(0,0,t)$ and control $q(0,0,t)$ over time at the origin of the spatial domain Ω for several shooting iterations. The initial control is chosen as $q_0 \equiv 0$ and from the second shooting iteration the control obeys the imposed constraints. Comparing these two figures to the corresponding Figures 6.2 and

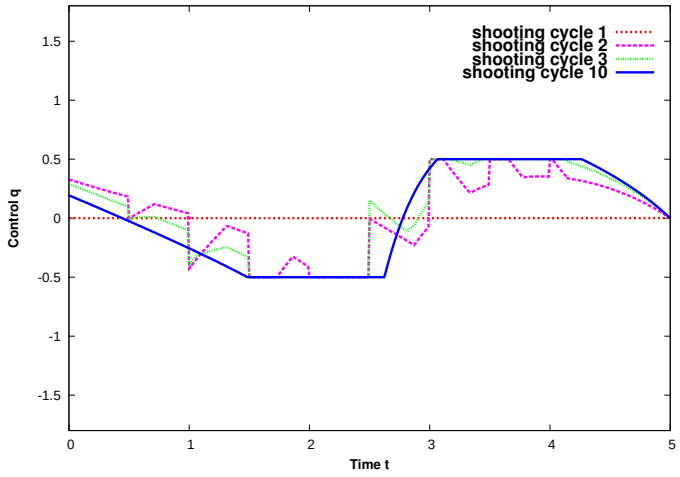

Figure 6.6. Example 6.1: The control $q(0,0,t)$ at different DMS cycles in the control constrained case for $\omega = 7$ and $\alpha = 0.5$.

6.3 shows the coincidence of IMS and DMS solutions after convergence.

For the IMS approach, Table 6.3 provides an argument why the choice of a minimum number of shooting intervals is sensible for solving a given problem. The same holds true in the DMS framework, as can be inferred from Table 6.5. Again, we see that the computational effort required for solving the shooting problem increases with a growing number of shooting intervals.

Table 6.5. Example 6.1: DMS for $\omega = 3$ and different equidistant shooting decompositions.

#SI	#Newton	#GMRES	$J(q,u)$	$\|F\|$	t(s)
1	10	10-12	2.208	$4.1{\cdot}10^{-06}$	1501
2	9	12-15	2.208	$3.1{\cdot}10^{-10}$	1588
4	8	15-19	2.208	$2.7{\cdot}10^{-11}$	1651
5	8	17-20	2.208	$2.7{\cdot}10^{-11}$	1730
10	7	22-28	2.208	$2.7{\cdot}10^{-11}$	1913
20	6	35-53	2.208	$2.7{\cdot}10^{-11}$	2702

This section concludes with an examination of the quality of solutions depending on the regularization parameter α. A similar proceeding in the unconstrained case (cf. Table 5.10 and Figure 5.7) reveals that a sequence of decreasing values of α leads to improved approximations of the minimum functional value and improves matchings of the tracking function $\hat{u}(x,t)$.

Table 6.6. Example 6.1: Dependence of the functional value on the regularization parameter α.

α	#Newton	$\#\text{GMRES}_{\text{min / max}}$	$\|F\|$	$J(q,u)$	t(s)
1	6	26/50	$1.5{\cdot}10^{-04}$	2.354	2599
0.5	9	28/50	$4.4{\cdot}10^{-04}$	2.195	4953
0.1	16	54/56	$7.3{\cdot}10^{-04}$	1.869	11413
0.05	26	66/74	$8.4{\cdot}10^{-04}$	1.798	25292

Table 6.6 illustrates that the optimal value of $J(q,u)$ decreases with the regularization parameter. However, small parameter values entail a significant increase of computing time required for solving the OCP.
Finally, Figures 6.7 and 6.8 depict the development of the state solution $u(0,0,t)$ and the control $q(0,0,t)$ at the center of the spatial domain Ω over time for different values of the regularization parameter α. The improvement in the state solution for decreasing α is smaller than in the corresponding unconstrained case (compare Figure 5.7). Moreover, the parts of the solution interval I where the control constraints are active increase with decreasing values of α.

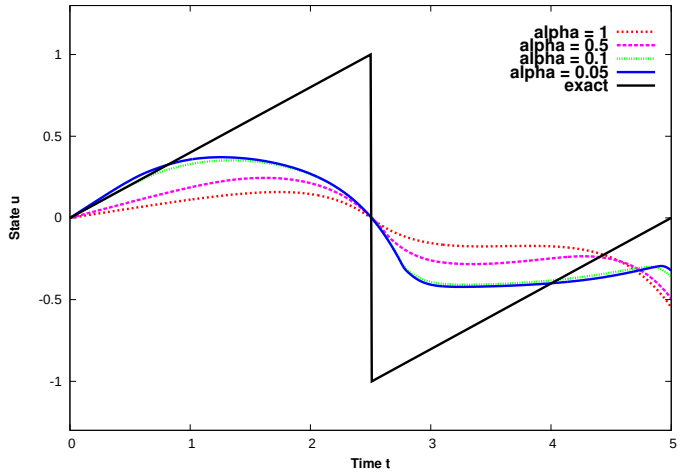

Figure 6.7. Example 6.1: Quality of the match of the state $u(0,0,t)$ to the tracking function $\hat{u}(0,0,t)$ depending on the regularization parameter α in the control constrained case.

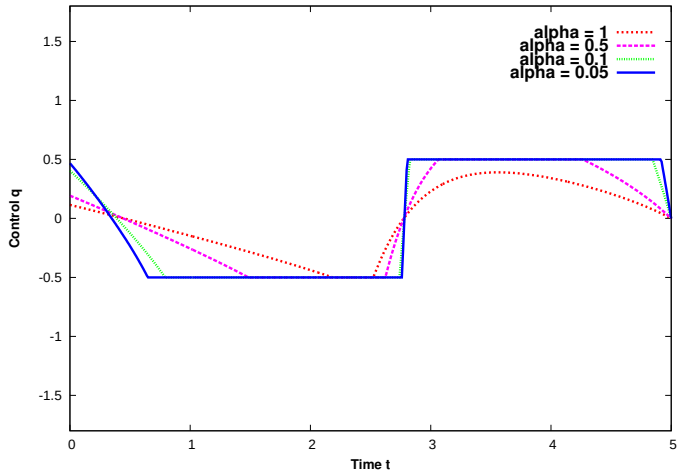

Figure 6.8. Example 6.1: The dependence of the control $q(0,0,t)$ on the regularization parameter α at different DMS cycles in the control constrained case.

6.4 Comparison of IMS and DMS

This section finally provides a comparison of direct and indirect shooting variants for PDE governed OCP. In the ODE optimal control framework, direct shooting methods are nowadays preferred over indirect ones, as they are more flexible in dealing with additional conditions such as control or state constraints which render control problems more complex. In the following, we set different perceptions of direct versus indirect methods in relation. Then the numerical results of Sections 5.5 and 6.3 are summarized.

In the ODE context direct shooting methods are often described as a 'first-discretize-then-optimize' concept. In the framework of classical DMS as described in Section 5.3, this means that in problem (5.25) both the controls q^j and the states $u^j(q^j, s^j)$ are discretized and the Lagrangian (5.27) is formulated on the discrete level. The resulting KKT system constitutes a large-scale nonlinear programming problem (NLP). So far, the proceeding is straightforward, but the solution of the NLP, e. g., by sequential quadratic programming (SQP) methods, requires delicate technical fine-tuning (see Potschka [94]).
However, the DMS concept is not necessarily based on an underlying discretization. There are problems, e. g., in ODE mixed integer programming, where discretizing the variables before the optimization process leads to erroneous results or even to unsolvable discrete problems. Thus, classical DMS can also be viewed as a 'first-optimize-then-discretize' approach, which corresponds to its presentation in Section 5.3. Our concept of DMS is derived from the same system of optimality conditions as IMS in Section 5.2. On the function space level, it constitutes a re-interpretation of classical DMS for PDE governed OCP.

In all numerical tests, the IMS and DMS approaches from Sections 5.1 and 5.2 are contrasted. We choose problem configurations for which simple shooting methods are unstable and a decomposition of the solution interval into several shooting subintervals is required.
In the following enumeration, we formulate some main conclusions drawn from the numerical results. They are confirmed by further similar tests not displayed in this work, which permits to generalize them for the considered problem classes.

1. Due to the often large shooting systems, employing a suitable preconditioner is important, particularly for fine shooting grids with a large number of shooting intervals. However, the results of a symmetric Gauss-Seidel preconditioner proposed by Comas [26], Heinkenschloss [50] and Hesse [52] remain unsatisfactory. The design of alternative preconditioners for shooting methods is crucial for future applications.

2. An increasing number of equidistant shooting intervals entails a growing number of Newton-GMRES iterations for the shooting system (cf. Table 5.6 as well as Tables 5.16 and 5.18). This leads to increasing CPU times for both IMS and DMS, see also Tables 6.3 and 6.5. The use of a global refinement strategy as proposed in Subsection 5.5.3 enhances the efficiency of both shooting approaches with respect to the number of GMRES iterations as well as to CPU time. Decreasing the regularization parameter α improves the tracking process triggered by the objective functional. These statements are valid both for unconstrained and control constrained examples.

3. Both IMS and DMS constitute two-step fixed-point iterations for the extended OCP (3.43) – (3.44), possibly complemented by the constraints (6.18). In IMS, the first step consists in the solution of subinterval BVP, which are replaced by intervalwise IVP in DMS. In both approaches, the second step is the solution of the respective shooting system. The discussion in Section 5.4, particularly the abstract example examined in Table 5.1, predicts larger shooting systems for DMS. As a consequence, the number of Newton-GMRES iterations in DMS is often twice as large as in IMS. This is confirmed by the corresponding tables in Sections 5.5 and 6.3.

4. In general, IMS and DMS provide equally good results with respect to convergence of the functional values and shooting residuals. For linear test examples, IMS often performs better than DMS, as displayed in the right panels of Tables 5.4 and 5.8 as well as in Table 5.11. However, for nonlinear examples the DMS method is far more efficient than IMS with respect to CPU time, although it requires almost twice as many iterations for solving the shooting system. This is confirmed by Tables 5.7 as compared to 5.9, as well as by Tables 5.12, 5.13, and 5.14. The superiority of DMS is further proven in the nonlinear case with control box constraints, as displayed in Tables 6.1 and 6.4. For the latter results, the performance is remarkable, as the DMS shooting system is solved by a damped Newton-GMRES method, which usually reduces the convergence rate of Newton's method.

In summary, the observations in Chapters 5 and 6 show that, although IMS performs better in solving simple linear examples, for complex problems direct shooting methods are often superior to indirect ones. This supports the preference of DMS in the ODE optimal control context over the past decades.

So far, all results were obtained on equidistant shooting grids. Several test examples both in the ODE and the PDE context suggest the use of an adaptive shooting technique. As there are no satisfactory adaptive multiple shooting methods in the literature, the next chapter is concerned with developing two novel approaches in this regard.
Indirect shooting methods for OCP are similar to the original shooting concept for BVP, as the corresponding splitting (5.1) of the optimality conditions leads to a set of intervalwise OCP that display a BVP structure. Therefore, the development of adaptive shooting techniques, which is first performed for ODE BVP, concentrates on IMS in the optimal control framework.

7 Adaptive Multiple Shooting

After completing the comparative investigation of direct and indirect shooting methods, this final chapter brings the topic of adaptivity into focus. The importance of handling the choice of shooting intervals in a problem-oriented manner has occurred several times before in our work (see, in particular, Chapters 2, 5 and 6).
First, we clarify the understanding of an adaptive shooting method. When adaptivity is connected to multiple shooting in the literature, this is mostly a matter of adapting the temporal or spatial discretization within the shooting intervals (for a recent example, see Hesse & Kanschat [53]). Distributing the shooting points themselves adaptively is, on the other hand, rarely considered. These approaches employ techniques developed independently from shooting methods, e. g., they compute shooting solutions on a fixed shooting grid but with adaptive mesh refinement on the single shooting intervals. In this chapter, we elaborate two approaches toward adaptively modifying the shooting grid. Up to now, little is known about how to design adaptive multiple shooting methods in the latter sense. Hence, we extend previous work by Mattheij & Staarink [83]. Our second approach is connected to an idea by Maier [80].
The mentioned work of Mattheij et al. discusses a way of distributing the shooting points adaptively due to the needs of the given problem. However, the authors only consider linear ODE boundary value problems, and there occur difficulties in transferring their approach to nonlinear problems. We summarize their ideas in Section 7.1 and develop an extension to nonlinear BVP and OCP in Section 7.2. Therefore, we return to problem classes from Chapter 2. In Section 7.3 our approach is applied to OCP governed by parabolic PDE, both without and with control box constraints. We thoroughly discuss the additional difficulties occurring in the PDE context. In each section we present numerical results within the respective problem classes which have been implemented in `MATLAB` or `deal.ii` and all results were achieved on the same computer.

7.1 Optimal choice of shooting intervals (SI) for linear BVP – the bounding approach

The general linear BVP reads as follows:

$$\begin{aligned} \dot{u}(t) &= A(t)u(t) + b(t), \quad t \in [a,b] \\ B_a u(a) + B_b u(b) &= g. \end{aligned} \tag{7.1}$$

Again, $A(\cdot) : I \to \mathbb{R}^{d\times d}$ and $b(\cdot) : I \to \mathbb{R}^d$ are continuous real-valued matrix and vector functions, $B_a,\ B_b \in \mathbb{R}^{d\times d}$ are given constant matrices, and $g \in \mathbb{R}^d$ is a given vector.

As shown in Chapter 2, solving such BVP with shooting methods, i. e., turning the BVP into an IVP with parameterized initial values, often invokes problems concerning the stability of the solution method. This holds even when the original BVP is well-conditioned and follows from stability estimates which depend exponentially on the length of the solution interval I (cf. (2.14)). A decomposition of I, leading from simple to multiple shooting, renders the solution process stable, because the size of the exponential stability factor depending on the interval length can then be controlled.
In this section, the fixed equidistant shooting grids are replaced by adaptively chosen grids. The discussion of Example 2.1 in Section 2.2 showed that the a priori chosen number of (equidistant) shooting intervals influences both the subinterval sensitivities $G^i(t)$ and the shooting matrix $F_s'(\bar{s})$ (see Table 2.2). This confirms that the number and distribution of shooting intervals is connected to the conditioning of the problem and to the stability of the shooting algorithm, as the sensitivities play a major role in BVP conditioning (see (2.10)). Assuming that the influence is mutual, it is suggestive to base the shooting point distribution on a control of the sensitivities $G^i(t)$. We recapitulate the multiple shooting system explicitly, thus recalling the required notation:

$$\begin{pmatrix} G^0 & -I & 0 & \cdots & 0 \\ 0 & G^1 & -I & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & G^{M-2} & -I \\ A & 0 & \cdots & 0 & B \end{pmatrix} \begin{pmatrix} \delta s^0 \\ \delta s^1 \\ \vdots \\ \delta s^{M-2} \\ \delta s^{M-1} \end{pmatrix} = - \begin{pmatrix} u^0(\tau_1; s^0) - s^1 \\ u^1(\tau_2; s^1) - s^2 \\ \vdots \\ u^{M-2}(\tau_{M-1}; s^{M-2}) - s^{M-1} \\ r(s^0, u^{M-1}(\tau_M; s^{M-1})) \end{pmatrix} \tag{7.2}$$

For the linear problem (7.1), the matrix blocks in the last row of (7.2) are given by $A = B_a$ and $B = B_b G^{M-1}$. It is crucial for the following that the sensitivities $G^i(t)$ are independent of the shooting variables s^i in the case of a linear BVP.
The basic proposition of Mattheij & Staarink (see [83]) is to bound the growth of the sensitivities in some matrix norm $\|\cdot\|$ by a constant C_{sens}. This is why we call the resulting adaptive shooting scheme the bounding approach. The idea originally arose from analyzing the global error ε accumulated during the solution process. After discussing different error contributions, Mattheij & Staarink suggested the following upper bound for the sensitivity growth:

$$\|G^i(t)\| \leq C_{\text{sens}} \lesssim \frac{\varepsilon}{\kappa M \epsilon_{\text{mach}}}. \tag{7.3}$$

Here, ε is the global error, κ describes the conditioning of the BVP (see (2.12)), M is the number of shooting intervals and $\epsilon_{\text{mach}} \approx 10^{-16}$ is the machine precision. Note that, if the problem at hand exhibits an exponential dichotomy, the factor M can be suppressed (see Mattheij [82]). The choice of shooting points is then performed as follows: During a simultaneous forward solution of the parameterized IVP (2.3) and the belonging sensitivity equation (2.7), whenever the norm $G(t_j)$ surmounts the prescribed constant C_{sens}, the current timepoint t_j is chosen as a new shooting point τ_i. Then the solution process is restarted with a parameter s^i as the new initial value. After the shooting intervals have been fixed, the shooting system (7.2) is solved. We test this proceeding by means of an example taken from the original article [83].

Remark 7.1. For all numerical examples in Sections 7.1 and 7.2, the Crank-Nicolson method is chosen as an ODE integrator (see Subsection 4.1.1). Both equidistant and adaptive computations are always carried out on a discretization of the respective solution intervals with 10000 timesteps.

Example 7.1. *The following BVP corresponds to Example 2.3 in Mattheij & Staarink [83] and is solved on the time interval* $I = [0, \pi]$.

$$\begin{pmatrix} \dot{u}^1(t) \\ \dot{u}^2(t) \\ \dot{u}^3(t) \end{pmatrix} = \begin{pmatrix} 1 - 19\cos(2t) & 0 & 1 + 19\sin(2t) \\ 0 & 19 & 0 \\ -1 + 19\sin(2t) & 0 & 1 + 19\cos(2t) \end{pmatrix} \begin{pmatrix} u^1(t) \\ u^2(t) \\ u^3(t) \end{pmatrix} - \begin{pmatrix} 2 - 19\cos(2t) + 19\sin(2t) \\ 19 \\ 19\cos(2t) + 19\sin(2t) \end{pmatrix}$$

The boundary matrices B_0 *and* B_π *are both given by the* 3×3 *identity matrix. The exact solution is given by the componentwise constant function* $u(t) \equiv (1, 1, 1)^\top$.

This problem shows an exponentially dichotomic behavior, hence, in (7.3) the factor M can be neglected. Furthermore, the BVP is well-conditioned, i. e., $\kappa \approx 1$. Therefore, equation (7.3) yields an upper bound $C_{\text{sens}} \lesssim 10^{16}\varepsilon$. The global error tolerance is $\varepsilon = 10^{-8}$, which finally requires $C_{\text{sens}} \leq 10^8$. In Table 7.1, we obtain a reproduction of the results from the

Table 7.1. Example 7.1: Number of shooting intervals depending on the spectral norm of the sensitivity matrices (stopping criterion for the shooting residual: $\|F(s)\|_2 < 10^{-8}$).

C_{sens}	10^1	10^2	10^3	10^4	10^5	10^6
$\|(u - u_h)(\pi)\|_2$	$1.6{\cdot}10^{-15}$	$5.5{\cdot}10^{-15}$	$1.4{\cdot}10^{-14}$	$1.0{\cdot}10^{-11}$	$4.8{\cdot}10^{-13}$	$7.9{\cdot}10^{-13}$
$\|F(s)\|_2$	$1.9{\cdot}10^{-14}$	$7.0{\cdot}10^{-14}$	$8.3{\cdot}10^{-12}$	$3.0{\cdot}10^{-11}$	$3.0{\cdot}10^{-10}$	$4.0{\cdot}10^{-09}$
#SI	28	14	10	7	6	5

original article concerning the correlation between the number of shooting intervals and the sensitivity size measured in the spectral norm.

Table 7.2. Example 7.1: Position of the shooting points obtained by bounding the sensitivity growth; length of the resulting shooting intervals as distance between consecutive points.

	$\|G(t)\|_2 \leq 10^4$			$\|G(t)\|_2 \leq 10^6$	
shooting point	position	distance	shooting point	position	distance
1	0	0.4593	1	0	0.6886
2	0.4593	0.4593	2	0.6886	0.6886
3	0.9186	0.4593	3	1.3773	0.6886
4	1.3779	0.4593	4	2.0659	0.6886
5	1.8372	0.4593	5	2.7545	0.3870
6	2.2965	0.4593	6	3.1416	
7	2.7558	0.3858			
8	3.1416				

As expected for a linear problem, the norm $\|F(s)\|_2$ of the shooting residual falls below the prescribed tolerance ε within one shooting iteration. Table 7.2 illustrates that the choice of shooting intervals leads to an equidistant shooting point distribution. An exception is the last interval being shorter than the previous ones. We conjecture that this equidistant shooting point distribution results from the uniform boundedness of the entries of the BVP's system matrix. In Example 7.1, all entries are bounded by the constant $K = 20$ on the time interval $I = [0, \pi]$. A similar behavior is expected for autonomous BVP, where the system matrices have constant entries. But this does not hold if the system matrix coefficients vary strongly over time or cannot be uniformly bounded. We confirm this issue by further examples below.
Although Example 7.1 supports the upper bound (7.3) postulated by Mattheij & Staarink, there are several critical matters of discussion:

1. The presence of the number M of shooting intervals is contradictory in an estimate which should enable an adaptive choice of shooting points. M is not known a priori, but it is the objective to find an appropriate number of shooting intervals.

2. Example 7.1 shows that the upper bound for C_{sens} depends proportionally on the prescribed global error bound ε, i. e., decreasing ε leads to equally decreasing sensitivity bounds. This appears questionable for linear problems, where the first shooting iteration already drives error components near the machine precision, independently of the number of shooting intervals. Table 7.1 shows that choosing ε within the interval $[10^{-9}, 10^{-4}]$ leads to different values of C_{sens} but does not change the results.

3. The dependence on the condition number κ of the continuous BVP indicated in (7.3) is not realistic, as the discussion of Example 7.2 below illustrates.

Even though it is plausible to postulate a relation between the conditioning of the problem, the stability of the shooting algorithm, and the adaptive optimal choice of shooting points, these objections clearly question an upper bound for the sensitivity constant C_{sens} in the spirit of (7.3). Lacking a confirmable alternative criterion, in the following examples we choose sensitivity constants $C_{\text{sens}} \in [10^3, 10^7]$. The adaptive process described and applied in this chapter therefore remains partly heuristic. As a basis for further examinations, the proceeding suggested by Mattheij & Staarink is summarized in Algorithm 7.1. Steps 6–11 represent standard multiple shooting known from Algorithm 2.1.

In the following, Example 2.1 is reconsidered. The aim is twofold: The example justifies skipping the bounding criterion (7.3), and it confirms the conjecture on adaptively detected equidistant shooting grids.

Example 7.2. *Consider the linear BVP*

$$\begin{pmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ c & 1 \end{pmatrix} \begin{pmatrix} u^1(t) \\ u^2(t) \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^1(0) \\ u^2(0) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1(10) \\ u^2(10) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

As in Section 2.2, increasing the value of the parameter c entails both ill-conditioning of the BVP and instability of the shooting algorithm. For $c = 110$, the condition number κ from

Algorithm 7.1 The approach of Mattheij & Staarink [83] in algorithmic form.

Require: Shooting variable s, bound C_{sens}

Prescribe tolerance TOL
Solve the initial value problem $\dot{u}(t) = A(t)u(t) + b(t)$, $u(a) = s$ and the variational equation $\dot{G}(t) = A(t)G(t)$, $G(a) = I_d$
if $\|G(t_j)\| > C_{\text{sens}}$ **then**
    Take the timepoint t_j as shooting point τ_i, restart solving with $u(\tau_i) = s$, $G(\tau_i) = I_d$
end if
while $\|F(\overline{s})\| > \text{TOL}$ **do**
    Solve the subinterval IVP and evaluate the residual $-F(\overline{s})$
    Solve the subinterval variational IVP and evaluate $F_s'(\overline{s})$
    Solve shooting system $F_s'(\overline{s})\delta\overline{s} = -F(\overline{s})$
    Compute update $\overline{s}_{\text{new}} = \overline{s} + \delta\overline{s}$, resolve the subinterval IVP
end while

(2.12), which is the one proposed by Mattheij [82], is given as $\kappa \approx 10^{44}$. Thus, the criterion (7.3) requires that $C_{\text{sens}} \lesssim \frac{10^{-8}}{10^{44} \cdot 10^{-16}} = 10^{-36}$ for a chosen tolerance $\varepsilon = 10^{-8}$ for the global error. This is impossible to achieve. Referring back to Table 2.1, we see that $M \approx 8$ shooting intervals are the optimal choice, which is obtained by choosing $C_{\text{sens}} \approx 10^6 - 10^7$. Furthermore, the system matrix in Example 7.2 has constant entries regardless of the value of the parameter c. According to our above conjecture, Algorithm 7.1 should result in equidistant shooting grids (only the last shooting interval may be shorter, as it contains the remainder that is left when dividing the length of the solution interval by the length of the detected adaptive shooting intervals). This behavior is confirmed by Figure 7.1. It is obvious that more shooting intervals are required for larger values of c. This matches the results of Section 2.2. Note also that, in case $c = 1$, the adaptive process results in simple shooting.

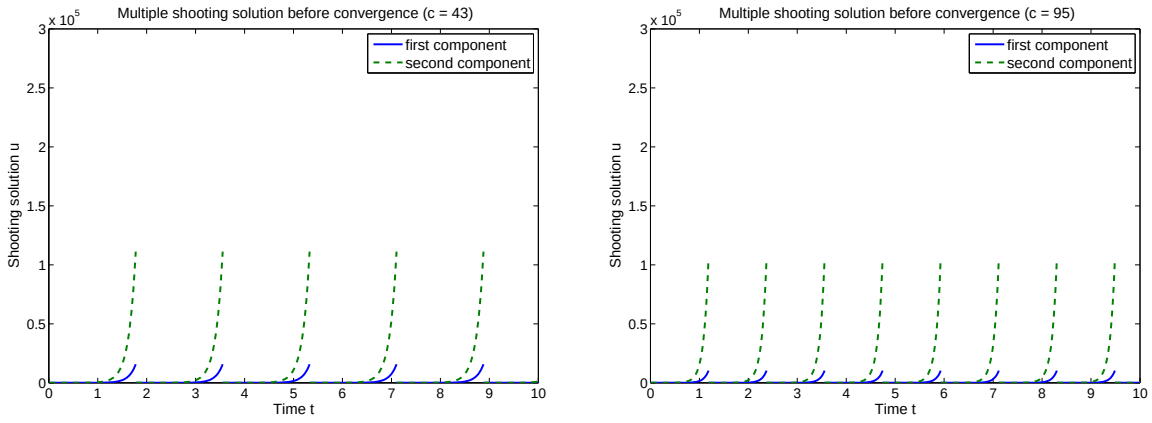

Figure 7.1. Example 7.2: The equidistant shooting grid found by Algorithm 7.1 (left: $c = 43$, right: $c = 95$); the last shooting interval comprises the remainder of the solution interval and is therefore shorter.

The observations presented in the current section substantiate the close connection between the number of shooting intervals and the 'size' of the sensitivity matrices that was postulated in Section 2.2 (see Table 2.1). So far, it is not clear how to measure this size best and we always bounded the spectral norm of the sensitivities, i. e., $\|G(t)\|_2$. In the following, two variants of Example 7.2 are considered to test several approaches to measure the sensitivities.
In the first variant of Example 7.2, the system matrix entries are piecewise constant on the solution interval and therefore still uniformly bounded. However, their size varies strongly over time and they exhibit discontinuities. Besides testing different strategies to measuring the size of $G(t)$ and choosing the shooting points, we observe a nonequidistant shooting point distribution.

Example 7.3. *Consider once again the BVP*

$$\begin{pmatrix}\dot{u}_1(t)\\ \dot{u}_2(t)\end{pmatrix} = \begin{pmatrix}0 & 1\\ c & 1\end{pmatrix}\begin{pmatrix}u^1(t)\\ u^2(t)\end{pmatrix}, \qquad \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}u^1(0)\\ u^2(0)\end{pmatrix} + \begin{pmatrix}0 & 0\\ 1 & 0\end{pmatrix}\begin{pmatrix}u^1(10)\\ u^2(10)\end{pmatrix} = \begin{pmatrix}1\\ 1\end{pmatrix}$$

on the time interval $I = [0, 10]$. The matrix entry c is now chosen as a piecewise constant function:

$$c(t) = \begin{cases}110 & (t \le 2 \ \vee \ t \ge 8),\\ 1 & (2 < t < 8).\end{cases}$$

Table 7.3 presents the results obtained by testing different quantities measuring the size of the sensitivities $G(t)$. As in Table 7.1 above different upper bounds for the sensitivities are tested (here, $C_{\text{sens}} \in [10^3, 10^6]$). The growth of $G(t)$ is bounded in the most common matrix norms, i. e., the $\|\cdot\|_1, \|\cdot\|_2$ and $\|\cdot\|_\infty$ norms, as well as in the spectral radius $\rho(\cdot)$, which is computed from the eigenvalue with largest modulus and is, in general, not a norm. The table illustrates that all quantities work almost equally well for determining the shooting grid with the spectral radius criterion resulting in less shooting points but simultaneously yielding a slightly larger shooting residual.

Table 7.3. Example 7.3: Number of shooting intervals and shooting residual for different sensitivity bounds and different strategies for measuring the sensitivity size (stopping criterion for the shooting residual: $\|F(s)\|_2 < 10^{-8}$).

C_{sens}	10^3		10^4		10^5		10^6	
strategy	#SI	$\|F\|_2$	#SI	$\|F\|_2$	#SI	$\|F\|_2$	#SI	$\|F\|_2$
$\|G\|_1$	11	$5.8 \cdot 10^{-13}$	7	$1.3 \cdot 10^{-12}$	6	$1.8 \cdot 10^{-11}$	5	$8.8 \cdot 10^{-11}$
$\|G\|_2$	11	$2.0 \cdot 10^{-13}$	7	$5.5 \cdot 10^{-12}$	6	$5.1 \cdot 10^{-11}$	5	$6.1 \cdot 10^{-10}$
$\|G\|_\infty$	11	$6.0 \cdot 10^{-13}$	7	$2.2 \cdot 10^{-12}$	6	$1.6 \cdot 10^{-11}$	5	$6.4 \cdot 10^{-10}$
$\rho(G)$	9	$2.2 \cdot 10^{-12}$	7	$3.8 \cdot 10^{-11}$	5	$3.1 \cdot 10^{-10}$	5	$1.7 \cdot 10^{-09}$

Next, in Table 7.4 we take a closer look at the shooting point distibutions resulting from the different quantities. The first three columns show that the choice of different matrix

norms leads to almost the same distributions; there are several shooting points in the subintervals $[0, 2]$ and $[8, 10]$ where the parameter $c = 110$, but no additional shooting point occurs in the center where $c = 1$. The choice of the spectral radius, however, results in a different distribution as there is an additional shooting point in the center. Otherwise, the distribution is similar. In all cases, the first two shooting intervals are of equal length, and the second-to-last interval is also equally long.

Table 7.4. Example 7.3: Position of the shooting points obtained by bounding the sensitivity growth; comparison of different strategies for measuring the sensitivity size.

shooting point	$\|G(t)\|_1 \leq 10^4$	$\|G(t)\|_2 \leq 10^4$	$\|G(t)\|_\infty \leq 10^4$	$\rho(G(t)) \leq 10^4$
1	0	0	0	0
2	0.6760	0.6830	0.6750	0.8330
3	1.3520	1.3660	1.3500	1.6660
4	2.1610	2.3850	2.2280	4.7560
5	7.7520	8.0020	7.8190	8.2620
6	8.6690	8.6850	8.6550	9.0950
7	9.3450	9.3680	9.3300	9.9280
8	10	10	10	10

Finally, Figure 7.2 shows the adapted shooting grid in the case $\rho(G(t)) \leq 10^4$, corresponding to the last column of Table 7.4, as well as the continuous solution obtained after one shooting iteration.

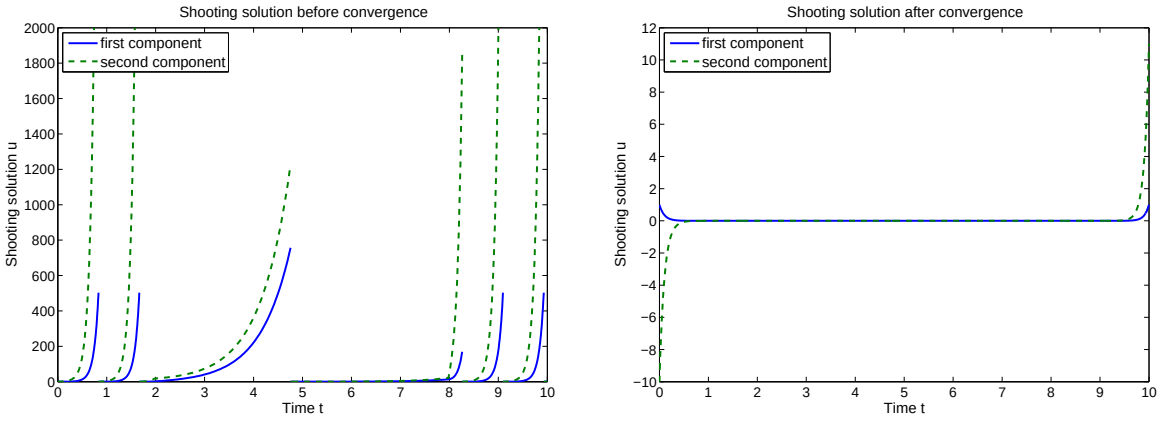

Figure 7.2. Example 7.3: The shooting solution found by Algorithm 7.1 (left: before convergence, right: after convergence); shooting points are mainly in regions where c is large (criterion: $\rho(G(t)) \leq 10^4$).

The last problem is a further variant of Example 7.2, where the parameter c is replaced by a linear function $c(t) = 11t$.

Example 7.4. *The BVP to be solved reads*

$$\begin{pmatrix} \dot{u}_1(t) \\ \dot{u}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 11t & 1 \end{pmatrix} \begin{pmatrix} u^1(t) \\ u^2(t) \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^1(0) \\ u^2(0) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1(10) \\ u^2(10) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

and the solution interval is again $I = [0, 10]$*. The exact solution is given by*

$$u^1(t) = c_1 e^{\frac{t}{2}} \mathrm{Ai}(a(t)) + c_2 e^{\frac{t}{2}} \mathrm{Bi}(a(t)),$$
$$u^2(t) = c_1 \left(\frac{1}{2} e^{\frac{t}{2}} \mathrm{Ai}(a(t)) + \sqrt[3]{11} e^{\frac{t}{2}} \mathrm{Ai}'(a(t)) \right) + c_2 \left(\frac{1}{2} e^{\frac{t}{2}} \mathrm{Bi}(a(t)) + \sqrt[3]{11} e^{\frac{t}{2}} \mathrm{Bi}'(a(t)) \right)$$

where $a(t) = \frac{11t + \frac{1}{4}}{\sqrt[3]{11}^2}$*,* c_1, c_2 *are constants, and* $\mathrm{Ai}(\cdot), \mathrm{Ai}'(\cdot)$ *and* $\mathrm{Bi}(\cdot), \mathrm{Bi}'(\cdot)$ *are the so-called Airy functions and their first derivatives. We note that* $\mathrm{Ai}(\cdot)$ *and* $\mathrm{Bi}(\cdot)$ *are linearly independent solutions of the Airy differential equation*

$$u''(t) - tu(t) = 0;$$

they occur in different areas of physics like optics, electromagnetics and quantum mechanics. For more information, see the compendium by Olver [91].

As in Example 7.3 above, we first test several ways of bounding the sensitivity growth. Table 7.5 displays the results for Example 7.4, which confirm that any of the chosen matrix norms as well as the spectral radius are suitable quantities for measuring the sensitivity size. As before, the spectral radius yields a minor improvement w. r. t. the number of shooting intervals.

Table 7.5. Example 7.4: Number of shooting intervals and shooting residuals for different sensitivity bounds and different strategies for measuring the sensitivity size.

C_{sens}	10^3		10^4		10^5		10^6	
strategy	#SI	$\|F\|_2$	#SI	$\|F\|_2$	#SI	$\|F\|_2$	#SI	$\|F\|_2$
$\|G\|_1$	14	$3.0 \cdot 10^{-13}$	10	$4.0 \cdot 10^{-12}$	8	$1.5 \cdot 10^{-11}$	7	$1.2 \cdot 10^{-10}$
$\|G\|_2$	14	$4.0 \cdot 10^{-13}$	10	$2.1 \cdot 10^{-12}$	8	$5.0 \cdot 10^{-11}$	6	$6.8 \cdot 10^{-10}$
$\|G\|_\infty$	14	$2.1 \cdot 10^{-13}$	10	$4.0 \cdot 10^{-12}$	8	$2.6 \cdot 10^{-11}$	7	$1.3 \cdot 10^{-10}$
$\rho(G)$	11	$6.9 \cdot 10^{-13}$	9	$3.8 \cdot 10^{-12}$	7	$6.7 \cdot 10^{-10}$	6	$6.8 \cdot 10^{-10}$

In Example 7.4, the system matrix entries are no longer uniformly bounded, but increase linearly with the time variable. Therefore, the shooting intervals should become smaller with increasing time. In Figure 7.3, this assumption is confirmed both for the spectral norm $\|G(t)\|_2$ and the spectral radius $\rho(G(t))$ in case that $C_{\mathrm{sens}} = 10^5$.

So far, the determination of shooting points was aligned with the size of the sensitivities, which is based on the ideas by Mattheij [82] and Mattheij & Staarink [83]. This proceeding is motivated by a theoretical connection between the conditioning of the given BVP, the stability of the shooting algorithm, and the sensitivity growth. Although this connection

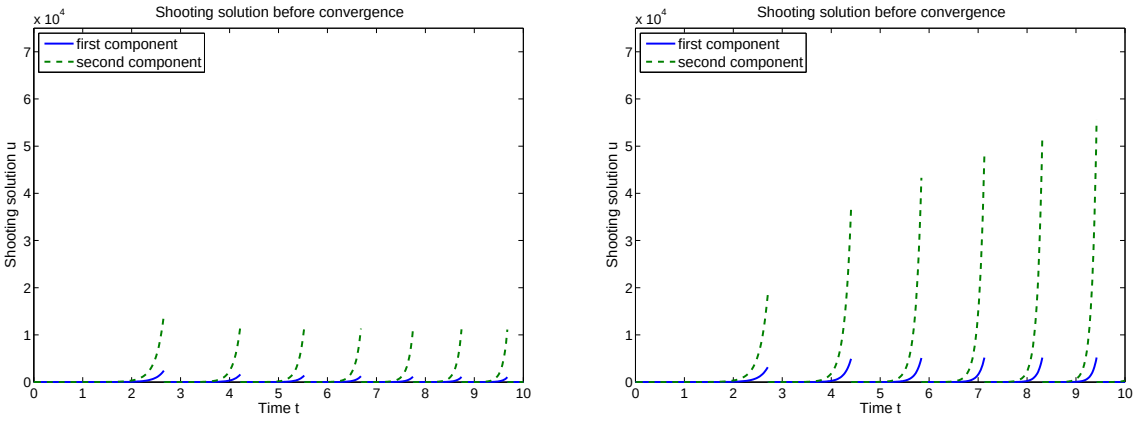

Figure 7.3. Example 7.4: The shooting solution found by Algorithm 7.1 (left: $\|G(t)\|_2 \leq 10^5$, right: $\rho(G(t)) \leq 10^5$).

is observed in examples (see Section 2.2), it is difficult to quantify it appropriately. A different and simpler approach consists in bounding the shooting solution itself. As Figures 7.1 – 7.3 illustrate, the subinterval shooting solutions u^1 and u^2 increase exponentially over time, meaning that the exponential growth behavior is not restricted to the sensitivities. Table 7.6 compares the numbers of shooting intervals and the shooting residuals detected by the adaptive Algorithm 7.1 based on bounding the solution, $\|u(t)\|_2$, or the sensitivities, $\|G(t)\|_2$, in the Euclidean respectively the spectral norm. The results suggest that bounding the solution u to determine the shooting grid works equally well as bounding the sensitivity growth. It even results in a slightly smaller number of adaptively detected shooting intervals. This behavior has been confirmed by several further examples, among them Examples 7.1 – 7.3.

Table 7.6. Example 7.4: Number of shooting intervals and shooting residual for different bounding strategies, i. e., $\|u(t)\|_2 \leq C_{\text{sens}}$ respectively $\|G(t)\|_2 \leq C_{\text{sens}}$, and different bounds.

C_{sens}	10^3		10^4		10^5		10^6	
strategy	#SI	$\|F\|_2$	#SI	$\|F\|_2$	#SI	$\|F\|_2$	#SI	$\|F\|_2$
$\|u\|_2$	10	$2.9 \cdot 10^{-12}$	8	$1.6 \cdot 10^{-11}$	7	$1.1 \cdot 10^{-10}$	6	$2.5 \cdot 10^{-09}$
$\|G\|_2$	14	$4.0 \cdot 10^{-13}$	10	$2.1 \cdot 10^{-12}$	8	$5.0 \cdot 10^{-11}$	6	$6.8 \cdot 10^{-10}$

Table 7.7 reveals that the shooting grids achieved by the two different approaches are very close to one another. The distribution displayed in the first column results from bounding the solution of the original problem, whereas the one in the third column relies on the solution of the variational equation. We presume that the similarity is grounded in the identical structure of both problems which is due to the linearity of the original BVP. We will test the approach of bounding $\|u(t)\|_2$ in Section 7.2 for nonlinear examples, where we expect its performance to deteriorate.

Table 7.7. Example 7.4: Position of the shooting points obtained by bounding the growth of the solution resp. of the sensitivities; length of the resulting shooting intervals as distance between consecutive points.

	$\|u(t)\|_2 \leq 10^5$			$\|G(t)\|_2 \leq 10^5$	
shooting point	position	distance	shooting point	position	distance
1	0	2.9770	1	0	2.6590
2	2.9770	1.7770	2	2.6590	1.5610
3	4.7540	1.4880	3	4.2200	1.3040
4	6.2420	1.3260	4	5.5240	1.1600
5	7.5680	1.2160	5	6.6840	1.0630
6	8.7840	1.1350	6	7.7470	0.9920
7	9.9190	0.0810	7	8.7390	0.9360
8	10		8	9.6750	0.3250
			9	10	

Each of the criteria introduced in this section will be re-examined in the nonlinear context, see Subsection 7.2.1.

In conclusion, a brief overview summarizes the advantages and drawbacks of the described adaptive approach for choosing the shooting points in linear BVP. The essential advantages in comparison with a fixed equidistant shooting grid are:

1. Given an upper bound C_{sens} for the growth of the sensitivities $\|G(t)\|$, the shooting intervals are chosen automatically.

2. If C_{sens} is chosen adequately, the minimum number of shooting points necessary for solving the given BVP as well as the corresponding optimal choice of shooting intervals is detected.

3. Whenever possible, Algorithm 7.1 leads to simple shooting. This was observed, e. g., for the case $c = 1$ in Example 7.2.

However, there are several disadvantages of the adaptive procedure displayed in the following list:

1. Due to the independence of $\|G(t)\|$ on the shooting variable s, the adaptive process ends up with an equidistant shooting point distribution if the BVP system matrix has uniformly bounded entries.

2. There exists no rule by which to choose the sensitivity bound C_{sens}, which renders the adaptive process partly heuristic.

3. In nonlinear problems, the sensitivities depend on the shooting variables s, i. e., $G(t) = G(t; s)$. This entails a dependence of the shooting grid on s, and therefore Algorithm 7.1 is not applicable to nonlinear BVP.

4. Algorithm 7.1 runs contrary to the potential of multiple shooting for computing the subinterval solutions in parallel. In order to detect the shooting points, the BVP has to be solved forward sequentially. Only afterwards multiple shooting can be applied in parallel on the sequentially determined shooting grid.

As we neglect the parallelization property of multiple shooting, we are mainly concerned with transferring the adaptive procedure to the nonlinear case. This is the main objective of the next section.

Remark 7.2. A different approach to adaptively distributing the shooting points was introduced by Maier [80] for singularly perturbed BVP. It is designed for handling nonlinear problems and is independent of stability theory, which allows to avoid heuristic criteria. It is not meaningful for linear problems as they usually converge in one shooting iteration and is therefore postponed to Subsection 7.2.2. There, we propose a concept similar to Maier's that is applicable to general ODE problems.

7.2 Optimal choice of SI for nonlinear BVP and ODE governed OCP

The strategy of Mattheij & Staarink [83] for adaptively determining the shooting grid discussed in the last section remains unsatisfactory in the following regard: As the solution process in linear examples converges within one shooting iteration, no genuine grid adaptation can be observed. The shooting grid is once fixed according to a given criterion, but afterwards not altered.
The current section deals with more advanced nonlinear problems. In Subsection 7.2.1, we are concerned with transferring Algorithm 7.1 to the nonlinear case. It turns out that additional precautions are necessary, which is justified in several steps. After the algorithm is adapted to nonlinear problems, several numerical examples illustrate its performance. Among them are nonlinear problems with linear boundary conditions, fully nonlinear BVP (i. e., both ODE and boundary conditions are nonlinear) and nonlinear OCP.
In Subsection 7.2.2, an alternative and independent approach toward adaptivity is discussed. Similar to a method proposed by Maier [80] for singularly perturbed BVP, the shooting process is started with a given very finely resolved shooting grid. In the following iterations, the grid is successively thinned out, i. e., unnecessary shooting intervals are removed. We also consider the possibility of inserting additional shooting points where this is necessary.

7.2.1 Extension of the bounding approach to nonlinear BVP

The approach to automatic shooting point distribution from Algorithm 7.1 cannot be expected to work for nonlinear BVP. This is plausible from the following observation. The

sensitivity or variational equation of the parameterized IVP

$$\dot{u}(t;s) = A(t)u(t;s) + b(t), \quad u(a;s) = s$$

(corresponding to the linear BVP (7.1)) is given by

$$\dot{G}(t) = A(t)G(t), \quad G(a) = I_d.$$

This sensitivity equation does not depend on the shooting parameter s, and therefore the multiple shooting grid detected by Algorithm 7.1, relying on bounding the sensitivities $G(t)$, is also independent of s. Furthermore, in the linear setting, only one shooting iteration has to be carried out, resulting in a fixed shooting grid. In contrast, the general nonlinear parameterized IVP,

$$\dot{u}(t;s) = f(t, u(t;s)), \quad u(a;s) = s,$$

leads to the sensitivity equation

$$\dot{G}(t;s) = f'_x(t, u(t;s))G(t;s), \quad G(a;s) = I_d.$$

The Jacobian $f'_x(t,x)$ depends on s, which then transfers to the sensitivity solution $G(t) \equiv G(t;s)$. Thus, the sensitivity bounding approach leads to shooting grids depending on s. We expect a different shooting grid in each iteration as in each shooting cycle, the parameter vector $\overline{s} = (s^0, s^1, \ldots, s^{M-1})$ is updated. Therefore, the approach from Section 7.1 must be modified in order to allow for changing grids. It is briefly summarized in the following pseudo-algorithm.

1: Distribute shooting points due to Mattheij's criterion of bounding the growth of $\|G(t)\|$
2: **while** $\|F(s)\| > TOL$ **do**
3: Solve the multiple shooting problem
4: **end while**

The obvious modification takes the distribution step inside the loop, so that the shooting points are redistributed in each shooting iteration. The current shooting grid is then determined by bounding the sensitivities that are computed on the basis of the respective last update of $\overline{s}$. Instead of fixed quantities that were sufficient in the equidistant framework, the implementation now has to deal with quantities of varying size for the shooting points, shooting variables, shooting solutions etc. The modified pseudo-algorithm reads

1: **while** $\|F(s)\| > TOL$ **do**
2: Distribute shooting points based on bounding the growth of $\|G(t;s)\|$
3: Carry out one multiple shooting iteration
4: **end while**

The problem after modification is illustrated in Figure 7.4 in a simplified way. Assume that in the i-th shooting iteration, we have shooting variables $\overline{s}_i = (s_i^0, s_i^1, \ldots, s_i^{M_i-1})$ and a corresponding shooting grid $\mathcal{T}_i$ (the upper part of the figure). Now one multiple shooting iteration is carried out, resulting in an update $\overline{s}_{i+1} = (s_{i+1}^0, s_{i+1}^1, \ldots, s_{i+1}^{M_i-1})$ of the shooting

variables. Based on $\overline{s}_{i+1}$, a new shooting grid $\mathcal{T}_{i+1}$ is determined, on which the next multiple shooting iteration must be carried out. Here the following problem arises: The solution of the original IVP as well as the sensitivity equations must be computed on the new grid $\mathcal{T}_{i+1}$, but as initial values only $\overline{s}_{i+1}$ are available, which have been computed on the old grid $\mathcal{T}_i$ (see the lower part of Figure 7.4). Indeed, the modified algorithm does not

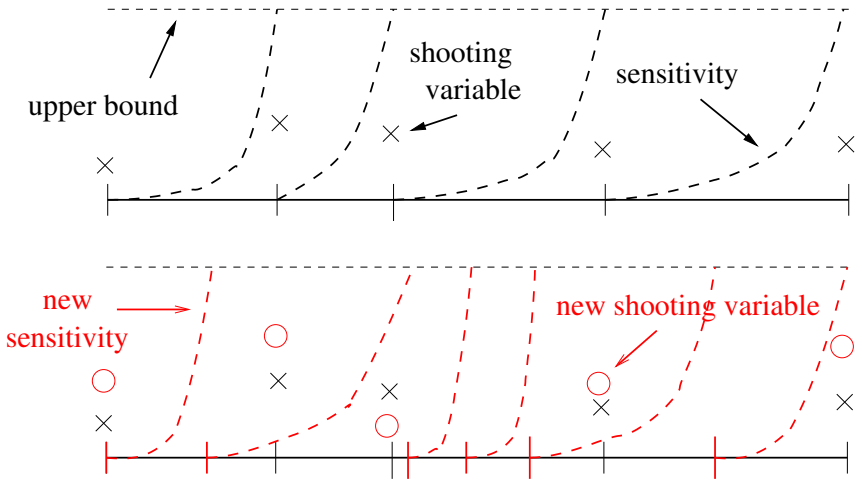

Figure 7.4. The adaptive multiple shooting process; upper part: shooting variables s (denoted by '×') and corresponding sensitivity growth (curved dashed lines); lower part: new shooting variables (denoted by '○') and new sensitivity norms (curved dashed lines). Problem: In the lower part, shooting variables and intervals do not fit.

work in practice. In most cases the quantities comprising the shooting points $\tau^{(i+1)}$ and the shooting variables $\overline{s}_{i+1}$ are of different length, because in the new shooting grid $\mathcal{T}_{i+1}$, either shooting points from $\mathcal{T}_i$ have been removed or additional ones have been inserted. Before carrying out a new multiple shooting iteration, the shooting variables $\overline{s}_{i+1}$ have to be matched to the new grid $\mathcal{T}_{i+1}$. The obvious way to achieve this matching is to interpolate

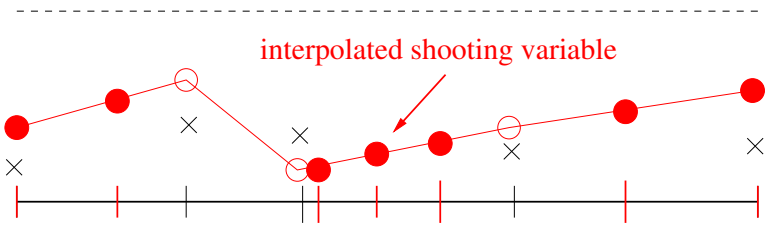

Figure 7.5. Interpolation of the shooting variables $\overline{s}_{i+1}$ (here: piecewise linear interpolation) and evaluation of the interpolant in the shooting points of the new grid $\mathcal{T}_{i+1}$ provides appropriate initial values for the multiple shooting iteration (denoted by '•').

the shooting values $\overline{s}_{i+1}$ and evaluate the interpolant at the gridpoints of $\mathcal{T}_{i+1}$, yielding suitable initial values $\overline{s}^{\text{int}}_{i+1}$ for the solution of original IVP. Figure 7.5 illustrates this idea

with piecewise linear interpolation. We continue to employ this in the examples below because it has turned out to work well in practice. Altogether, the modified procedure can be written in the following pseudo-algorithmic form:

1: **while** $\|F(s)\| > TOL$ **do**
2: Distribute shooting points based on bounding the growth of $\|G(t;s)\|$
3: Interpolate the shooting variables and take the values of the interpolant as new initial values on the subintervals
4: Carry out one multiple shooting iteration
5: **end while**

The steps enabling the transfer of the adaptive bounding approach to nonlinear BVP are presented in detail in the following Algorithm 7.2. The additional features can be easily included into existing implementations of multiple shooting algorithms. For OCP, the modification of indirect shooting methods is straightforward and is reconsidered below.

Algorithm 7.2 The bounding approach to adaptive multiple shooting for nonlinear problems (an extension of Algorithm 7.1).

Require: Shooting variable s, bound C_{sens}
1: Prescribe tolerance TOL
2: **while** $\|F(\overline{s})\| >$ TOL **do**
3: Solve the initial value problem $\dot{u}(t;s) = f(t;u(t;s))$, $u(a;s) = s$ and the variational equation $\dot{G}(t;s) = f'_x(t;u(t;s))G(t;s)$, $G(a;s) = I_d$
4: **if** $\|G(t_j;s)\| > C_{\text{sens}}$ **then**
5: Take the timepoint t_j as shooting point τ_i
6: Compute a piecewise interpolant y of the shooting variables $\overline{s}$
7: Restart solving with $u(\tau_i;s) = s$, $G(\tau_i;s) = I_d$ in the first iteration, $u(\tau_i;s) = y(\tau_i)$, $G(\tau_i;s) = I_d$ in subsequent iterations (where y is the interpolant from step 6)
8: **end if**
9: Solve the subinterval IVP and evaluate the residual $-F(\overline{s})$
10: Solve the subinterval variational IVP and evaluate $F'_s(\overline{s})$
11: Solve shooting system $F'_s(\overline{s})\delta\overline{s} = -F(\overline{s})$
12: Compute update $\overline{s}_{\text{new}} = \overline{s} + \delta\overline{s}$
13: **end while**

The remainder of this subsection presents two examples for this adaptive shooting method. First, a nonlinear BVP is considered; besides the functionality of the adaptive mechanism, an influence on the domain of convergence of Newton's method for the shooting system is observed. The second example is a nonlinear OCP which is adaptively solved by IMS (as introduced in Section 2.3). The different criteria for determining the shooting points discussed in Section 7.1 are tested in the nonlinear framework. To the best of our knowledge there exists no similar adaptive mechanism in the literature.

Example 7.5. *Consider the nonlinear BVP*

$$\ddot{u}(t) = \frac{3}{2}u(t)^2, \quad u(0) = 4,\, u(10) = \frac{4}{121}$$

with linear boundary conditions on the interval $[0, 10]$*. This problem is a modification of a BVP taken from Bulirsch & Stoer [20]. Its exact solution is given by* $u(t) = \frac{4}{(1+t)^2}$*. We rewrite it as a two-component first order BVP in the form*

$$\begin{pmatrix} \dot{u}^1(t) \\ \dot{u}^2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{3}{2}u^1(t)^2 \\ u^2(t) \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^1(0) \\ u^2(0) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1(10) \\ u^2(10) \end{pmatrix} = \begin{pmatrix} 4 \\ \frac{4}{121} \end{pmatrix}.$$

In Table 7.8, the adaptive development of the multiple shooting process for Example 7.5 is illustrated. The number of shooting intervals needed per iteration successively decreases until only two shooting intervals are required in the last few shooting cycles. From the last column, an increase in convergence order can be observed at the end of the solution process.

Table 7.8. Example 7.5: Number of shooting intervals in different shooting cycles, minimum and maximum length $|I_j|$ of the detected shooting intervals, and size $\|F\|_2$ of the corresponding shooting residual (initial shooting variable $s = (-9, \sqrt{23})^\top$, criteria $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$).

iteration	#SI	$\min \lvert I_j \rvert$	$\max \lvert I_j \rvert$	$\|F\|_2$
5	47	0.1950	0.6070	$8.5 \cdot 10^{04}$
10	27	0.1300	0.9970	$1.1 \cdot 10^{04}$
15	15	0.4840	1.5290	$1.6 \cdot 10^{03}$
20	9	0.2120	2.1080	$2.2 \cdot 10^{02}$
25	5	1.5370	2.4750	$2.6 \cdot 10^{01}$
30	3	2.5950	4.0670	$2.0 \cdot 10^{00}$
35	2	2.6130	7.3870	$5.0 \cdot 10^{-02}$
38	2	2.6140	7.3860	$5.8 \cdot 10^{-10}$

Figure 7.6 depicts the adaptive behavior of the multiple shooting solver. Shooting intervals of different length as well as changing shooting grids are clearly recognizable. Note also the labelling of the y axes indicating the improvement of the shooting solutions. Table 7.9 presents a comparison of the equidistant multiple shooting method and the adaptive procedure for two different pairs of initial values. The criterion for adapting the shooting grid is to bound the spectral radius of the sensitivity matrix by $C_{\text{sens}} = 10^2$. It is a general observation confirmed by several examples that C_{sens} has to be chosen smaller in the nonlinear framework than for linear problems. Summarizing the results of Table 7.9, the adaptive process is not only faster than equidistant multiple shooting, it also takes less shooting intervals, and it converges for initial parameters s for which equidistant shooting fails completely. For $s = (-9, \sqrt{23})^\top$, the inner Newton method for solving the implicit time steps needs more than $\max_{\text{it}} = 500$ iterations; the inner Newton method is

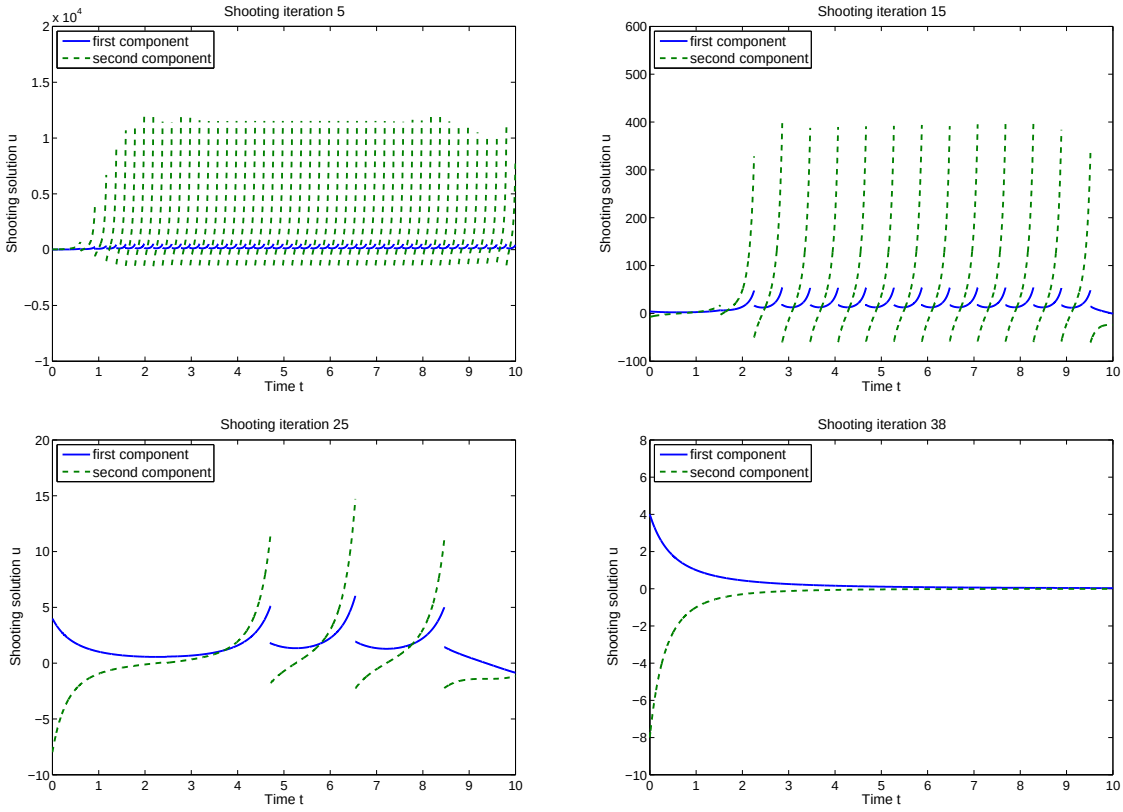

Figure 7.6. Example 7.5: The shooting solution found by the modified adaptive procedure (upper left: 5^{th} shooting cycle, upper right: 15^{th} shooting cycle, lower left: 25^{th} shooting cycle, lower right: 38^{th} shooting cycle (convergence)); the adaptive process successively reduces the number of shooting intervals (initial shooting variable $s = (-9, \sqrt{23})^\top$, criteria: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$).

then stopped, and the subinterval solutions cannot be computed. The latter observation suggests a positive impact of the adaptive shooting process on the domain of convergence of Newton's method. Although a thorough analysis of the domain of convergence is a hard task and has not been carried out, Figure 7.7 confirms this conjecture. The domain of convergence of the adaptive process (displayed in the left panel, where the observation is restricted to integer pairs in the domain $[-5, 5]^2$) comprises the upper left quadrant of the coordinate system, for which the equidistant shooting method (with 100 shooting intervals) fails to converge. A similar behavior has been observed with several nonlinear examples.

Remark 7.3. We emphasize that the domains of convergence for equidistant and adaptive multiple shooting depicted in Figure 7.7 have been obtained for Newton's method without any damping strategy. A suitably chosen damping parameter will presumably enlarge the convergence domain of Newton's method, both for equidistant shooting and for the

Table 7.9. Example 7.5: Comparison of equidistant and adaptive shooting for different initial shooting variables. Adaptive shooting is faster and has a larger convergence domain (criteria: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$). The notation '–(x)' means that the inner Newton method for the implicit ODE solver failed in shooting step x, needing more than 500 Newton iterates.

	$s = (0.1, -0.1)^\top$			$s = (-9, \sqrt{23})^\top$		
strategy	#SI	#Newton	time(s)	#SI	#Newton	time(s)
equidistant	5	–(3)	–	5	–(2)	–
	10	–(4)	–	10	–(3)	–
	20	>100	>128	20	–(4)	–
	50	>100	>129	50	–(6)	–
	100	>100	>130	100	–(6)	–
	200	>100	>130	200	–(7)	–
	500	>100	>134	500	–(8)	–
adaptive	2–9	24	62	2–58	38	102

sensitivity bounding strategy. However, the quadratic convergence observed in the last Newton iterate usually deteriorates if Newton's method is damped.

Finally, in Table 7.10, the different strategies for distributing the shooting points discussed in Section 7.1 are compared in the nonlinear context. Bounding different sensitivity norms

Table 7.10. Example 7.5: Number of shooting intervals and iterations, shooting residual and computing time for different strategies of measuring the sensitivity size (data: $s = (-9, \sqrt{23})^\top, C_{\text{sens}} = 10^2, \|F\|_2 \leq 10^{-8}$).

strategy	#SI	#Newton	$\|F\|_2$	time(s)
$\|G\|_1$	2-112	30	$2.0 \cdot 10^{-11}$	77
$\|G\|_2$	2-99	30	$1.4 \cdot 10^{-09}$	79
$\|G\|_\infty$	2-118	30	$1.8 \cdot 10^{-13}$	78
$\rho(G)$	2-58	38	$5.8 \cdot 10^{-10}$	102
$\|u\|_2$	1-22	58	$6.2 \cdot 10^{-12}$	164

yields equally good results, whereas the process becomes slightly slower when the spectral radius is bounded. Bounding the solution itself is not competitive in this example, and is expected to fail for nonlinear problems that are not finally solved on one shooting interval, as the solution u always grows on the first shooting interval until the bound C_{sens} is reached. We therefore exclude this strategy from further observations.
The second example in this subsection is a nonlinear OCP known from Chapter 2. It is part of the benchmark problem collection PROPT by Edvall & Rutquist [99] and is briefly repeated for the reader's convenience:

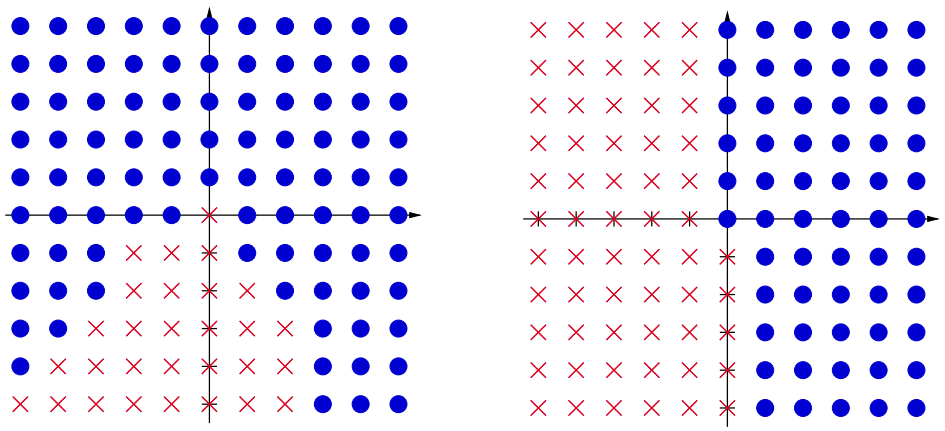

Figure 7.7. Example 7.5: The domains of convergence of Newton's method for the shooting system depending on the initial shooting value $s = (s_1, s_2)^\top$ (exemplarily the square $[-5, 5]^2$): adaptive multiple shooting (left), equidistant multiple shooting (right). The adaptive procedure has a larger domain of convergence ('•' means convergence, '×' means failure).

Example 7.6 (PROPT: Benchmark 51). *Consider the nonlinear optimization problem*

$$\begin{aligned} \min_{(q,u)} \quad J(q,u) \quad &= \quad \int_0^{10} (u^2(t) + q^2(t))\,\mathrm{d}t \\ s.\,t. \qquad \dot{u}(t) \quad &= \quad -u^3(t) + q(t), \quad u(0) = 1, \;\; u(10) = \frac{3}{2}. \end{aligned}$$

with an ODE boundary value problem as side condition. According to Rao & Mease [97], this problem is very sensitive to perturbations and therefore suited for multiple shooting solvers. The benchmark functional value is given as $J_{\min} = 6.723925$.

In Chapter 2 both IMS and DMS were applied to this problem. From a structural point of view, IMS is more similar to the shooting algorithm for BVP, as it exploits the BVP structure of the KKT system. In particular, the IMS Algorithm 2.2 requires the solution of an additional sensitivity equation, whereas in the DMS variant introduced in the ODE context the computation of sensitivities is concealed. In order to employ the adaptive shooting procedure without modification, we constrain ourselves to the indirect shooting variant in the following discussion. Table 7.11 shows a comparison between the equidistant and the adaptive multiple shooting method for Example 7.6.

Table 7.11. Example 7.6: Comparison of equidistant and adaptive multiple shooting for two different initial shooting variables (criteria: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$). The notation '–(x)' means that the inner Newton method for the implicit ODE solver failed in shooting step x, needing more than 500 Newton iterates.

	$s = (0.1, -0.1)^\top$				$s = (-25, -25)^\top$			
strategy	#SI	$\#_{\text{Newt}}$	$J(q,u)$	time(s)	#SI	$\#_{\text{Newt}}$	$J(q,u)$	time(s)
equidistant	5	–(2)	–	–	5	–(1)	–	–
	10	8	6.724	14	10	–(1)	–	–
	20	6	6.724	10	20	–(1)	–	–
	50	6	6.724	10	50	–(1)	–	–
	100	6	6.724	10	100	–(1)	–	–
	200	6	6.724	10	200	–(1)	–	–
	500	6	6.724	11	500	–(7)	–	–
	1000	6	6.724	13	1000	13	6.724	29
adaptive	3–10	33	6.750	85	3–770	64	6.750	172

As in the nonlinear BVP, we see that it is not possible to predict a priori whether the equidistant approach works for a given pair of initial shooting variables and a given fixed number of shooting intervals. In the case $s = (-25, -25)^\top$ displayed on the right, less than 500 equidistantly distributed shooting intervals lead to convergence failure. The adaptive shooting algorithm, on the other hand, finds the required number and distribution of shooting points during the computation. Although the adaptive process is slower than the equidistant approach, we emphasize that it runs automatically, whereas the equidistant counterpart is a trial and error approach. However, the optimal functional value is less well approximated by the adaptive shooting method, because the stopping criterion $\|F\|_2 < 10^{-8}$ is fulfilled before the minimum is reached.

Table 7.12. Example 7.6: Number of shooting intervals, functional value $J(q,u)$, and size $\|F\|_2$ of the corresponding shooting residual in different shooting cycles (initial shooting variable $s = (0.1, -0.1)^\top$, criteria $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$).

$\rho(G) \leq 10^1$				$\rho(G) \leq 10^2$			
iteration	#SI	$J(q,u)$	$\|F\|_2$	iteration	#SI	$J(q,u)$	$\|F\|_2$
3	8	13.48	$7.1 \cdot 10^{0}$	5	7	54.75	$4.7 \cdot 10^{1}$
6	7	12.14	$8.8 \cdot 10^{0}$	10	5	23.59	$2.5 \cdot 10^{1}$
9	6	8.27	$3.1 \cdot 10^{0}$	15	5	39.15	$5.0 \cdot 10^{1}$
12	6	6.85	$2.3 \cdot 10^{-1}$	20	4	20.31	$3.4 \cdot 10^{1}$
14	6	6.76	$6.6 \cdot 10^{-10}$	25	3	14.78	$1.9 \cdot 10^{1}$
				30	3	6.99	$4.5 \cdot 10^{-1}$
				33	3	6.75	$1.2 \cdot 10^{-13}$

In Table 7.12, details of the adaptive shooting solutions are illustrated for two different values of the sensitivity bound C_{sens}. We observe that the number of shooting intervals decreases in the course of the iteration, while the functional value and the shooting residual both converge. Again, a higher convergence order for the shooting residual is observed in the last iterations.
Finally, Figure 7.8 depicts the results of the left part of Table 7.12. Different shooting grids achieved by the successive iterations as well as nonequidistant shooting point distributions within the single grids are observed. The behavior of the presented adaptive algorithm will be shown in the context of further examples in the next subsection, where the results are compared to those obtained by a different adaptive approach.

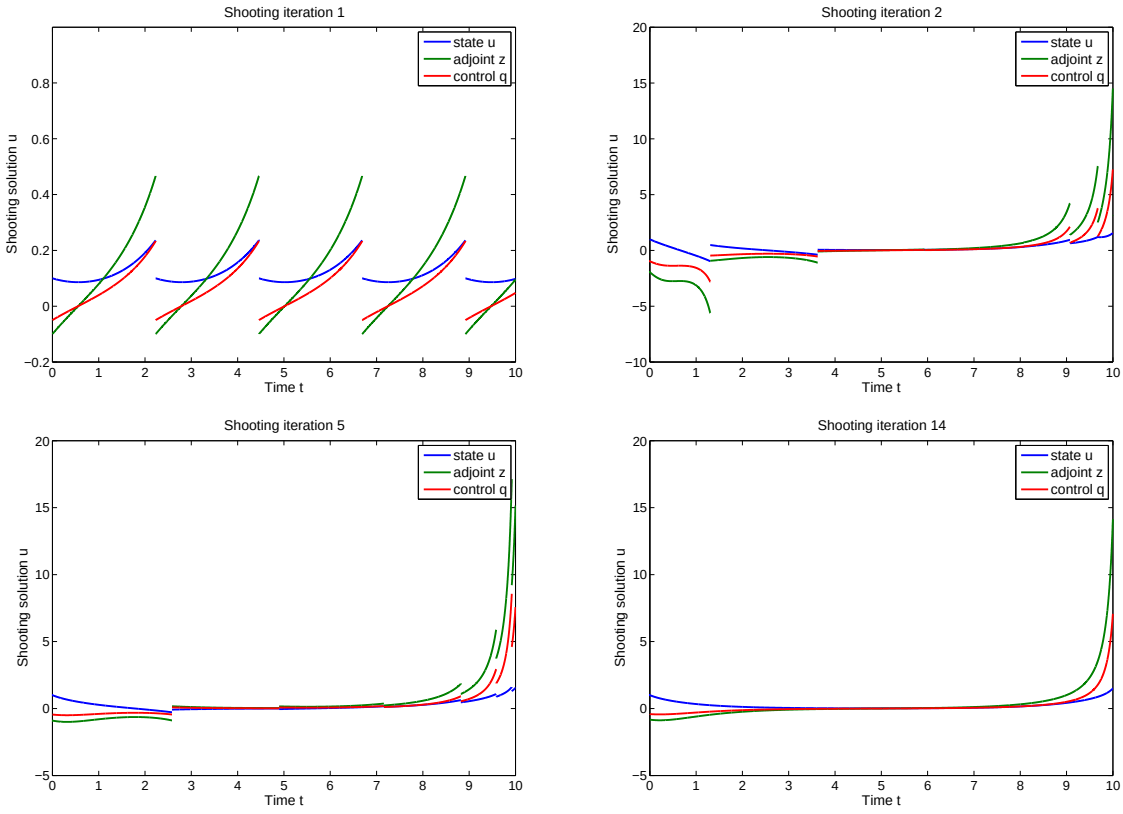

Figure 7.8. Example 7.6: The state u (blue), adjoint (green) and control (red) found by the adaptive procedure (upper left: 1^{st} shooting cycle, upper right: 2^{nd} shooting cycle, lower left: 5^{th} shooting cycle, lower right: 14^{th} shooting cycle (initial shooting variable $s = (0.1, -0.1)^\top$, criteria: $\rho(G(t)) \leq 10^1$, $\|F\|_2 < 10^{-8}$).

Remark 7.4. Although Table 7.10 suggests that bounding the spectral radius, $\rho(G) \leq C_{\text{sens}}$, is less efficient than bounding a sensitivity norm, we chose the spectral radius criterion for the problems in this and the subsequent subsections. If the adaptive bounding approach is transferred to PDE problems, bounding sensitivity norms is impossible as the sensitivity

matrices are not assembled. There are, however, matrix-free approaches (similar to the matrix-free Krylov-Newton methods discussed in Section 4.3) to approximate single eigenvalues of matrices that are not explicitly known. Thus, in the PDE framework the spectral radius criterion is more appropriate.

7.2.2 Successive reduction of the number of SI – the thinning approach

The main idea of this subsection is inspired by an adaptive multiple shooting method proposed by Maier [80]. The article is concerned with singularly perturbed BVP which depend on small parameters often leading to boundary layers of the solution. This means that the solution exhibits rapid variations near the interval boundaries but behaves otherwise regularly. Employing a multiple shooting method for solving such problems, one expects that near the boundaries many short shooting intervals are required, whereas the interior of the interval can be covered by one single shooting interval. Maier's approach consists in prescribing an initial shooting grid which accounts for this problem structure, i. e., there are initially several shooting intervals in the boundary layers but only few in the interior. In each shooting iteration, shooting points are inserted or removed according to certain estimates of eigenvalues of the Jacobian of the respective problem. As we have no particular interest in singularly perturbed BVP, we skip a detailed discussion of these eigenvalue criteria. Maier's approach lacks a theoretical justification but is shown to work in practical examples in [80].
The idea of adapting a given shooting grid step by step to the structure of the given problem inspired the following considerations. Starting with an initial grid (which is usually chosen equidistantly for lack of knowledge on the solution structure but could also incorporate given information), in each shooting iteration either additional shooting points are inserted or dispensable shooting points are removed. As the objective of multiple shooting is to achieve $F(\overline{s}) < \mathrm{TOL}$, it is suggestive to modify the shooting grid according to the size of local residual contributions. This leads to a residual-based adaptive process.
The further presentation is based on a shooting grid $\mathcal{T} = \{\tau_j\}_{j=0}^{M}$ and the corresponding old shooting variables $\overline{s} = (s^0, \dots, s^{M-1})^\top$ and update $\overline{s}_{\mathrm{new}} = (s^0_{\mathrm{new}}, \dots, s^{M-1}_{\mathrm{new}})^\top$. The shooting residual $F(\overline{s})$ consists of several components:

$$F(\overline{s}) = (F_1(s^0, s^1), F_2(s^1, s^2), \dots, F_{M-1}(s^{M-2}, s^{M-1}), F_M(s^0, s^{M-1}))^\top.$$

Each component describes either the jump in a shooting point or the prescribed boundary condition:

$$\begin{aligned} F_j(s^{j-1}, s^j) &= s^j - u_{j-1}(\tau_j, s^{j-1}), \quad (j = 1, \dots, M-1), \\ F_M(s^0, s^{M-1}) &= r(s^0, u_{M-1}(\tau_M, s^{M-1})). \end{aligned}$$

In order to obtain local residual quantities, we measure the norms $\|F_j(s^{j-1}, s^j)\|_2$, and for the grid adaptation process, we require the mean component size, F_{mean}, as well as the maximal distance between any two component norms, $F^{\max}_{\mathrm{dist}}$. They are given as follows:

$$F_{\mathrm{mean}} := \frac{\sum_{k=1}^{M} \|F_k\|_2}{M}, \quad F^{\max}_{\mathrm{dist}} := \max_j \|F_j\|_2 - \min_j \|F_j\|_2.$$

By means of these quantities, we modify the shooting grid as follows. If the local residual $\|F_j\|_2$ exceeds the mean component residual F_{mean} by more than a fixed fraction of $F_{\text{dist}}^{\max}$, i. e.,

$$\|F_j\|_2 \geq F_{\text{mean}} + \alpha_{\text{up}} F_{\text{dist}}^{\max},$$

then we insert an additional shooting point into the grid, namely $\tau_j^- = \frac{\tau_{j-1}+\tau_j}{2}$ which lies between the current and the previous one. Note that no additional point is inserted to the right of the current shooting point, as the size of the local residual depends only on previous timepoints but not on successive ones. If, on the other hand, $\|F_j\|_2$ falls below F_{mean} by more than a certain fraction of the maximal distance $F_{\text{dist}}^{\max}$, i. e.,

$$\|F_j\|_2 \leq F_{\text{mean}} - \alpha_{\text{low}} F_{\text{dist}}^{\max},$$

then we remove the corresponding shooting point τ_j from the grid. Residual components F_j with

$$F_{\text{mean}} - \alpha_{\text{low}} F_{\text{dist}}^{\max} \leq \|F_j\|_2 \leq F_{\text{mean}} + \alpha_{\text{up}} F_{\text{dist}}^{\max}$$

do not contribute to the grid adaptation. In the first case, additional shooting variables at the inserted grid points are required for the next shooting iteration. They are obtained by linear interpolation of the original neighboring shooting variables, i. e., $s_{\text{new}}^{j,-} = \frac{s_{\text{new}}^{j-1}+s_{\text{new}}^{j}}{2}$ at the new shooting point τ_j^-. In the second case, we remove the shooting variable s_{new}^j corresponding to the dispensable shooting point τ_j from the set of shooting variables.

Algorithm 7.3 Residual-based grid adaptation procedure within the thinning approach.

Require: Shooting grid $\mathcal{T}$, shooting variables $\overline{s}$, update $\overline{s}_{\text{new}}$ and coarsening and refinement constants α_{low} and α_{up}

1: Evaluate the norms $\|F_j\|_2$ of the local residual contributions for $j = 1, \ldots, M$
2: Compute the mean residual component size, F_{mean}, and the maximal distance between local residual components, $F_{\text{dist}}^{\max}$
3: **if** $\|F_j\|_2 \geq F_{\text{mean}} + \alpha_{\text{up}} F_{\text{dist}}^{\max}$ **then**
4: Insert the shooting point $\tau_j^- = \frac{\tau_{j-1}+\tau_j}{2}$ into the grid $\mathcal{T}$
5: **end if**
6: **if** $\|F_j\|_2 \leq F_{\text{mean}} - \alpha_{\text{low}} F_{\text{dist}}^{\max}$ **then**
7: Remove the shooting point τ_j from the grid $\mathcal{T}$
8: **end if**
9: For each inserted shooting point τ_j^-, prescribe the mean value $s_{\text{new}}^{j,-} = \frac{s_{\text{new}}^{j-1}+s_{\text{new}}^{j}}{2}$ of the neighboring shooting variables as additional shooting variables
10: For each removed shooting point τ_j, remove the corresponding shooting variable s_{new}^j from the set of shooting variables

Before summarizing the grid adaptation process in Algorithm 7.4, we discuss some peculiarities. First, the interval endpoints τ_0 and τ_M have to be treated separately. To avoid changing the solution interval, they cannot be removed from the shooting grid. Furthermore, at the left endpoint no additional shooting point can be inserted. These issues have to be accounted for in the implementation.
As the convergence of equidistant multiple shooting depends both on the initially chosen

shooting variables and on the number of equidistant shooting intervals prescribed at the beginning (see Tables 7.9 and 7.11 in the previous subsection as well as 7.14 below), it is advisable to start with a fine shooting grid of 50 to 500 shooting intervals. Hence, we expect the adaptive process to coarsen the shooting grid rather than further refine it. The examples below confirm that, after a possible shooting grid refinement in the first iterations, the coarsening prevails in the long run. Thus, shooting points are rather removed than inserted and the grid is thinned out. This is why we call the resulting residual-based adaptive shooting scheme the thinning approach. A variant of this thinning approach used later in the PDE context does not permit insertion of new points but only removal of current ones.
Algorithm 7.3 presents the adaptive process in an implementable manner. The complete adaptive shooting process is then given by Algorithm 7.4.

Algorithm 7.4 Thinning approach to adaptive multiple shooting for nonlinear problems.

Require: Initial decomposition $I = \{\tau_0\} \cup \bigcup_{j=0}^{M-1} (\tau_j, \tau_{j+1}]$, shooting variable $\overline{s}$
1: Prescribe tolerance TOL
2: **while** $\|F(\overline{s})\| >$ TOL **do**
3: Solve the subinterval IVP on the current shooting grid, evaluate the residual $-F(\overline{s})$
4: Solve the subinterval variational IVP on the current shooting grid, evaluate $F_s'(\overline{s})$
5: Solve shooting system $F_s'(\overline{s})\delta\overline{s} = -F(\overline{s})$
6: Compute update $\overline{s}_{\text{new}} = \overline{s} + \delta\overline{s}$
7: Adapt the shooting grid and the shooting variables according to Algorithm 7.3
8: **end while**

The first example in this subsection, Example 7.7, is a fully nonlinear BVP, i. e., both the differential equation and the boundary conditions are nonlinear. Besides testing the adaptive thinning approach, the problems are additionally solved by both the equidistant and the adaptive bounding approach from Subsection 7.2.1 to enable a comparison between all methods.

Example 7.7. *Consider the nonlinear BVP*

$$\dot{u}(t) = -4u(t)^{\frac{3}{2}} + 24t^2u(t)^2$$

on the interval $I = [0, 5]$ *together with the nonlinear boundary conditions*

$$\sin(u(0)) = \sin((1+\sqrt{2})^{-2}) \approx 1.707 \cdot 10^{-1}, \quad \sin(u(5)) = \sin((26+\sqrt{2})^{-2}) \approx 1.331 \cdot 10^{-3}.$$

One solution is given by $u(t) = (t^2 + 1 + \sqrt{2})^{-2}$ *but it is not unique due to the periodicity of the nonlinear boundary conditions. The numerical computations rely on the formulation*

$$\begin{pmatrix} \dot{u}^1(t) \\ \dot{u}^2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -4u^1(t)^{\frac{3}{2}} + 24t^2u^1(t)^2 \\ u^2(t) \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \sin(u^1(0)) \\ u^2(0) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \sin(u^1(5)) \\ u^2(5) \end{pmatrix} = \begin{pmatrix} 1.707 \cdot 10^{-1} \\ 1.331 \cdot 10^{-3} \end{pmatrix}.$$

Table 7.13. Example 7.7: Number of shooting intervals in different shooting cycles, minimum and maximum shooting interval length $|I_j|$, and size $\|F\|_2$ of the shooting residual (initial shooting variable $s = (0.8, 0.8)^\top$, criteria: $\alpha_{\text{low}} = \frac{1}{8}$, $\alpha_{\text{up}} = \frac{7}{8}$, $\|F\|_2 < 10^{-8}$).

iteration	#SI	$\min \lvert I_j \rvert$	$\max \lvert I_j \rvert$	$\|F\|_2$
1	100	0.05	0.05	$1.2 \cdot 10^{2}$
3	64	0.05	0.20	$9.8 \cdot 10^{0}$
5	42	0.05	0.80	$7.8 \cdot 10^{-1}$
7	36	0.05	1.60	$6.0 \cdot 10^{-2}$
9	34	0.05	2.20	$2.4 \cdot 10^{-3}$
11	27	0.05	2.55	$2.2 \cdot 10^{-5}$
13	20	0.05	3.55	$2.1 \cdot 10^{-10}$

Table 7.13 illustrates several features of the thinning approach. The solution process was started with $M = 100$ shooting intervals of equal length $|I_j| = 0.05$, and the grid was not further refined (the minimum interval length remains constant). Although for some nonlinear BVP, a refinement of the shooting grid is observed in the first shooting iterations, the final grids are always coarser than the initially chosen equidistant shooting point distribution. This depends on the initial number of shooting intervals.

Table 7.14. Example 7.7: Comparison of equidistant shooting and the two adaptive approaches for $s = (0.8, 0.8)^\top$ (criteria for the bounding approach: $\rho(G(t)) \leq 10^{1}$, $\|F\|_2 < 10^{-8}$; criteria for the thinning approach: $\alpha_{\text{low}} = \frac{1}{8}$, $\alpha_{\text{up}} = \frac{7}{8}$, $\|F\|_2 < 10^{-8}$). The notation '–(x)' means that the Newton method for the implicit ODE solver failed in shooting step x, needing more than 500 Newton iterates.

strategy	#SI	#Newton	$\|F\|_2$	time(s)
equidistant	5	–(1)	–	–
	10	–(1)	–	–
	20	–(1)	–	–
	50	15	$8.5 \cdot 10^{-14}$	25
	100	13	$1.8 \cdot 10^{-10}$	22
	200	13	$1.2 \cdot 10^{-10}$	22
	500	13	$7.6 \cdot 10^{-11}$	23
adaptive (bounding)	3–40	17	$1.6 \cdot 10^{-10}$	56
adaptive (thinning)	20–100	13	$2.1 \cdot 10^{-10}$	22

In Example 7.7, the total amount of shooting intervals is successively reduced. Similarly to the adaptive bounding approach (cf. Tables 7.8 and 7.12), an increased convergence order for the shooting residual norm $\|F\|_2$ is observed in the last solution steps, although the Newton system changes from one iteration to the next. In Table 7.14, the adaptive thinning approach is compared to different equidistant shooting grids as well as the adaptive

bounding approach.
Equidistant multiple shooting is as fast as the thinning method from Algorithm 7.4 but the latter provides a suitable sequence of adapted shooting grids automatically. In contrast, the equidistant approach fails when the grid is chosen too coarse. Compared to the adaptive thinning method, the bounding approach from Subsection 7.2.1 is significantly slower. We presume that this is due to the structure of the bounding approach, where in each shooting iteration the subinterval problems have to be solved twice: the first solution sweep detects the shooting points and cannot be performed in parallel, and the second solution sweep is required after the shooting values have been transferred from the old grid to the new one via interpolation. The thinning approach avoids this double solve by determining the new shooting grid and distributing appropriate shooting variables simultaneously.

Remark 7.5. The results from Table 7.14, while providing an argument for rejecting the bounding approach, can also make the thinning approach appear questionable. If there is no improvement in comparison to equidistant shooting, and as one has to also start on a fine shooting grid, it could be argued that there is no benefit in using the thinning approach instead of a fine equidistant shooting grid. However, in the PDE case presented in Section 7.3, the thinning approach is combined with the global space mesh refinement strategy from Subsection 5.5.3. In this context, it is shown to outperform equidistant multiple shooting.

Example 7.7 exhibits another interesting feature; as the boundary conditions are given by nonlinear periodic functions, the solution of the problem is not uniquely determined. This is confirmed by Figure 7.9; the solution achieved by multiple shooting depends on the initially chosen shooting variables.

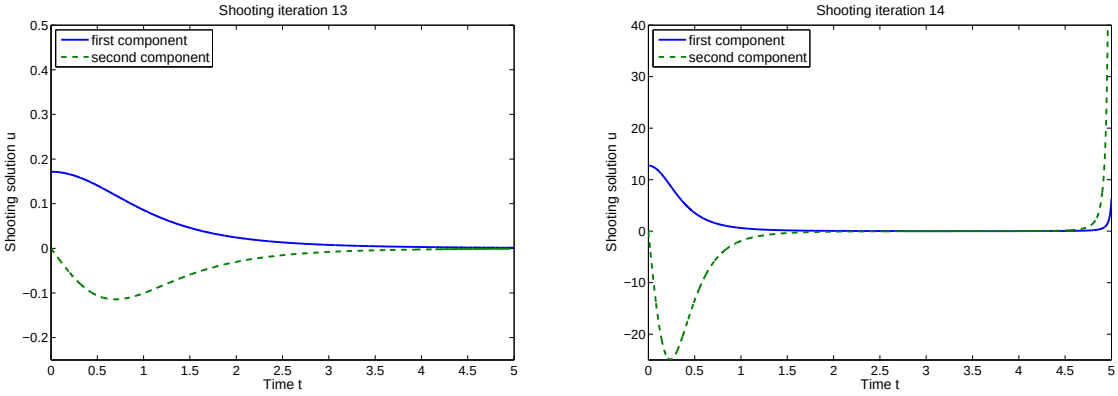

Figure 7.9. Example 7.7: Two different solutions; initial shooting value $s = (0.8, 0.8)^\top$ leading to $u^1(0) = 0.1715729, u^1(5) = 0.0013306$ and boundary values $\sin(u^1(0)) = 1.707 \cdot 10^{-1}, \sin(u^1(5)) = 1.331 \cdot 10^{-3}$ (left); initial shooting value $s = (8, -0.2)^\top$ leading to $u^1(0) = 12.7379435, u^1(5) = 6.2845159$ and boundary values $\sin(u^1(0)) = 1.707 \cdot 10^{-1}, \sin(u^1(5)) = 1.331 \cdot 10^{-3}$ (right).

Table 7.15. Example 7.5: Comparison of the two adaptive approaches. The adaptive thinning process takes more shooting intervals but is faster (criteria for the bounding approach: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$; for the thinning approach: $\alpha_{\text{low}} = \frac{1}{8}$, $\alpha_{\text{up}} = \frac{7}{8}$, $\|F\|_2 < 10^{-8}$).

strategy	#SI	#Newton	$\|F\|_2$	time(s)
bounding	2–58	38	$5.8 \cdot 10^{-10}$	102
thinning	18–100	11	$4.7 \cdot 10^{-10}$	13

Before moving to a second example, we revisit the examples from Subsection 7.2.1 and examine whether they exhibit the same behavior as depicted in Table 7.14. The results are summarized in Tables 7.15 and 7.16. In both cases, the thinning approach is considerably faster than the adaptive bounding approach. Table 7.16 further demonstrates that the thinning approach works also for OCP and yields comparably good results as the adaptive bounding approach from Subsection 7.2.1. Note that, in the OCP example, no grid refinement was allowed; instead, the adaptive process was concentrated on coarsening the shooting grid.

Table 7.16. Example 7.6: Comparison of the two adaptive approaches. The adaptive thinning process takes more shooting intervals but is faster (criteria for the bounding approach: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$; for the thinning approach: $\alpha_{\text{low}} = \frac{1}{8}$, $\|F\|_2 < 10^{-8}$). The thinning approach was carried out without grid refinement; therefore α_{up} is not specified.

strategy	#SI	#Newton	$J(q,u)$	$\|F\|_2$	time(s)
bounding	3–10	33	6.750	$1.2 \cdot 10^{-13}$	85
thinning	10–100	6	6.751	$5.3 \cdot 10^{-14}$	10

Example 7.8. *Consider the nonlinear BVP*

$$\dot{u}(t) = u(t)^2 + 2\pi^2 \cos(2\pi t) - \sin(\pi t)^4$$

on the interval $I = [0,4]$ *together with linear boundary conditions* $u(0) = 0$ *and* $u(4) = 0$. *The exact solution is given by* $u(t) = \sin(\pi t)^2$. *For the implementation, we reformulate the problem as a first order system*

$$\begin{pmatrix} \dot{u}^1(t) \\ \dot{u}^2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1(t)^2 \\ u^2(t) \end{pmatrix} + \begin{pmatrix} 2\pi^2 \cos(2\pi t) - \sin(\pi t)^4 \\ 0 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^1(0) \\ u^2(0) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1(4) \\ u^2(4) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

This second problem is chosen to illustrate that the adaptive thinning approach is superior to the bounding approach from Subsection 7.2.1. Up to now, several examples attested that

thinning is faster than bounding, which is again confirmed by Table 7.17. Furthermore, this table displays the maximum error $\max_t \|e(t)\|_2$ between the exact solution u_{ex} and the computed approximation u. On the given solution interval, the bounding approach yields $\max_t \|e(t)\|_2 \approx 2.1$, whereas the error obtained by the thinning approach with different initial shooting grids is smaller by about three orders of magnitude. The large error with

Table 7.17. Example 7.8: Comparison of the two adaptive approaches for the initial shooting value $s = (2, -2.5)^\top$. The adaptive thinning process is tested for $M \in \{10, 25, 100\}$ initial shooting intervals (criteria for the bounding approach: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$; for the thinning approach: $\alpha_{\text{low}} = \frac{1}{8}$, $\alpha_{\text{up}} = \frac{7}{8}$, $\|F\|_2 < 10^{-8}$).

strategy	#SI	#Newton	$\|F\|_2$	$\max_t \|e(t)\|_2$	time(s)
bounding	2–6	9	$8.9 \cdot 10^{-15}$	$2.1 \cdot 10^{0}$	11
thinning$_{10}$	1–10	7	$2.4 \cdot 10^{-13}$	$9.5 \cdot 10^{-3}$	5
thinning$_{25}$	3–25	6	$1.8 \cdot 10^{-09}$	$9.5 \cdot 10^{-3}$	4
thinning$_{100}$	38–100	6	$2.1 \cdot 10^{-10}$	$9.5 \cdot 10^{-3}$	4

the adaptive bounding method suggests a deviation from the exact solution, which is confirmed by Figure 7.10. On the other hand, the thinning solution is depicted in Figure 7.11. The solution after convergence of the multiple shooting process coincides with the exact solution. Finally, Table 7.18 shows that on the short solution interval $I = [0, 2]$, the two approaches yield equally good results. On the interval $I = [0, 10]$, however, the adaptive thinning approach is still able to solve the problem, whereas the bounding approach fails completely. However, we note that with increasing interval length, the thinning procedure takes more computing time. On $I = [0, 10]$ the results of Table 7.18 were achieved within 11 seconds.

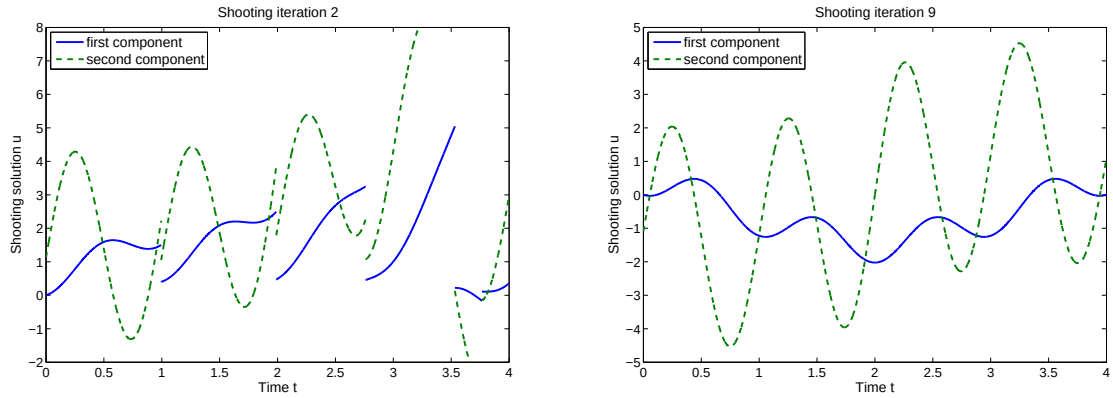

Figure 7.10. Example 7.8: The shooting solution found by the modified adaptive procedure from Subsection 7.2.1 (left: 2^{nd} shooting cycle, right: 9^{th} shooting cycle (convergence)); in this case, the adaptive process results in a wrong solution.

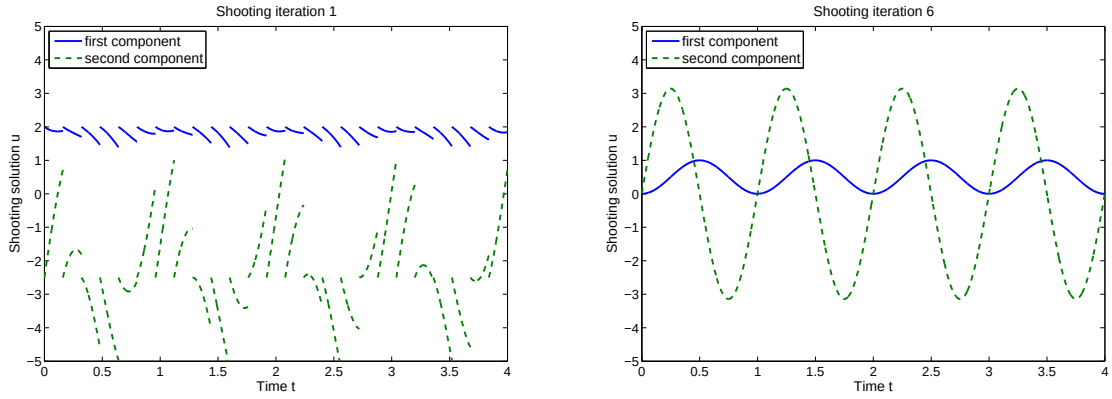

Figure 7.11. Example 7.8: The shooting solution found by the adaptive thinning procedure (left: 1^{st} shooting cycle, right: 6^{th} shooting cycle (convergence)); the thinning approach yields the correct solution.

Table 7.18. Example 7.8: Comparison of the two adaptive approaches for the initial shooting value $s = (2, -2.5)^\top$ and for several solution intervals of different length (criteria prescribed for the bounding approach: $\rho(G(t)) \leq 10^2$, $\|F\|_2 < 10^{-8}$; for the thinning approach: $\alpha_{\text{low}} = \frac{1}{8}$, $\alpha_{\text{up}} = \frac{7}{8}$, $\|F\|_2 < 10^{-8}$).

	$I = [0, 2]$		$I = [0, 4]$		$I = [0, 10]$	
strategy	#SI	$\max_t \|e(t)\|_2$	#SI	$\max_t \|e(t)\|_2$	#SI	$\max_t \|e(t)\|_2$
bounding	2–4	$1.1 \cdot 10^{-5}$	2–6	$2.1 \cdot 10^{0}$	–	–
thinning_{100}	42–100	$9.5 \cdot 10^{-3}$	38–100	$9.5 \cdot 10^{-3}$	15–100	$9.5 \cdot 10^{-3}$

7.3 Optimal choice of SI for parabolic OCP

Implementing the sensitivity based adaptive method from Subsection 7.2.1 for PDE examples, one is confronted with an additional problem. As provided in Chapter 5, the multiple shooting solvers in PDE optimal control are realized in a matrix-free manner. Therefore, the sensitivity matrices $G(t; s)$ on the subintervals are not available, i. e., their norm is difficult to evaluate. Therefore, we tested the spectral radius $\rho(G(t; s))$ of the sensitivities as a criterion for determining new shooting points in Subsection 7.2.1, as the computation of single eigenvalues of the sensitivities can be achieved in principle by means of a matrix-free Arnoldi algorithm. However, as the numerical tests in the ODE case revealed a superiority of the residual-based thinning over the sensitivity based bounding approach, the latter is not tested for PDE examples.
Instead, we concentrate on transferring the residual-based adaptive multiple shooting approach from Subsection 7.2.2 to the PDE optimal control framework. Its employment in the PDE context is straightforward. Nevertheless, in the practical realization a minor

modification is made. As we are interested in removing shooting points from the original grid in order to decrease the size of the shooting system, insertion of additional points is not permitted in the PDE case. This leads to the following Algorithm 7.5 which is a simpler variant of Algorithm 7.3:

Algorithm 7.5 Residual-based grid coarsening procedure within the thinning approach.

Require: Shooting grid $\mathcal{T}$, shooting variables $\overline{s}$, update $\overline{s}_{\text{new}}$, and coarsening constant α_{low}

1: Evaluate the norms $\|F_j\|_2$ of the local residual contributions for $j = 1, \ldots, M$
2: Compute the mean residual component size, F_{mean}, and the maximal distance between local residual components, $F_{\text{dist}}^{\max}$
3: **if** $\|F_j\|_2 \leq F_{\text{mean}} - \alpha_{\text{low}} F_{\text{dist}}^{\max}$ **then**
4: Remove the shooting point τ_j from the grid $\mathcal{T}$
5: **end if**
6: For each removed shooting point τ_j, remove the corresponding shooting variable s_{new}^j from the set of shooting variables

As before in the ODE case, α_{low} is a heuristically chosen parameter that influences the decision whether a shooting point is removed or not.
In this section, the adaptive process is started on an equidistant shooting grid comprising more shooting intervals than necessary for solving the problem. However, it cannot be chosen as fine as in the ODE case, as the conditioning of the shooting system deteriorates with an increasing number of shooting intervals. An initial shooting grid of 50 or more equidistant shooting intervals would require a suitable preconditioner. In Chapter 5, a symmetric Gauss-Seidel preconditioner was examined for linear problems. Although it was rejected due to its lack in efficiency, we found that for a large number of shooting points preconditioning is indispensable (see, e. g., Table 5.5). Thus, for the examples of this section, initial shooting grids with 10 equidistant subintervals are prescribed.
The goal of this section is twofold: first, we compare the adaptive shooting approach to the equidistant one, similar to the proceeding in Subsection 7.2.2. Second, it is verified whether employing the adaptive thinning approach in addition to global mesh refinement further improves the efficiency of IMS. The intention is to refine the spatial mesh while simultaneously removing shooting points that are not required.

Example 7.9. *Consider the problem*

$$\min_{(q,u)} J(q,u) = \frac{1}{2}\|u(x,T) - \hat{u}_T\|_{L^2(\Omega)}^2 + \frac{\alpha}{2}\int_0^T \|q(x,t)\|_{L^2(\Omega)}^2 \, dt$$

on $\Omega = (-1,1)^2$ *and with end-time* $T = 5$*, subject to the nonstationary nonlinear Helmholtz equation*

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + u(x,t)^3 &= q(x,t) \quad \text{in } \Omega \times I, \\ u(x,t) &= 0 \quad \text{on } \partial\Omega \times I, \\ u(x,0) &= u_0(x) \quad \text{in } \Omega. \end{aligned}$$

As multiple shooting converges within one single iteration for linear examples and the effect of the thinning approach is observable only over several iterations, two nonlinear problems are chosen in the following. In the first test case, Example 5.4 is revisited, where a given function at the final time has to be matched. In this example the heuristic shooting

Table 7.19. Example 7.9: IMS on 10 equidistant shooting intervals without (left) and with global refinement (right) as proposed in Subsection 5.5.3; number of GMRES iterations, development of functional values and shooting residual, CPU time (criterion: $\|F\|_2 < 5 \cdot 10^{-5}$).

	without refinement				with refinement				
iter.	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)	ref.	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)
0	–	2.37	$3.8 \cdot 10^{0}$	–	1	–	0.281	$1.2 \cdot 10^{0}$	–
1	51	0.117	$2.1 \cdot 10^{-1}$	1150	2	20	0.0658	$5.0 \cdot 10^{-1}$	13
2	28	0.115	$4.8 \cdot 10^{-3}$	1754	3	25	0.0918	$3.6 \cdot 10^{-1}$	67
3	28	0.115	$1.6 \cdot 10^{-4}$	2359	4	27	0.114	$2.6 \cdot 10^{-1}$	224
4	39	0.115	$4.8 \cdot 10^{-5}$	4310	4	28	0.115	$4.4 \cdot 10^{-4}$	829
5					4	39	0.115	$3.1 \cdot 10^{-5}$	2770

grid coarsening parameter is fixed at $\alpha_{\text{low}} = \frac{7}{8}$. Four different indirect shooting variants are contrasted. The plain IMS is complemented first by global space mesh refinement, then separately by the adaptive thinning mechanism, and finally by a combination of both. From the numerical results in Chapters 5 and 6 it is recommendable to use a spatial mesh refinement strategy, as it accelerates numerical computations significantly while obtaining equally good values for both objective functionals and shooting residuals. Table 7.19 provides the results of equidistant IMS on 10 shooting intervals both without and with global space mesh refinement. Although the solution with refinement requires an additional

Table 7.20. Example 7.9: Residual-based adaptive IMS without (left) and with global refinement (right) as proposed in Subsection 5.5.3; the refinement levels are the same as in Table 7.19. In addition, the number of shooting intervals (SI) is displayed (criterion: $\|F\|_2 < 5 \cdot 10^{-5}$).

	without refinement					with refinement				
iter.	$\#_{\text{SI}}$	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)	$\#_{\text{SI}}$	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)
0	10	–	2.37	$3.8 \cdot 10^{0}$	–	10	–	0.281	$1.2 \cdot 10^{0}$	–
1	10	51	0.117	$2.1 \cdot 10^{-1}$	1145	10	20	0.0658	$5.0 \cdot 10^{-1}$	14
2	10	28	0.115	$5.5 \cdot 10^{-3}$	1751	10	25	0.0918	$3.6 \cdot 10^{-1}$	68
3	8	24	0.115	$1.8 \cdot 10^{-4}$	2327	8	23	0.114	$2.6 \cdot 10^{-1}$	217
4	6	21	0.115	$2.1 \cdot 10^{-5}$	3699	8	24	0.115	$3.3 \cdot 10^{-4}$	787
5						8	24	0.115	$2.6 \cdot 10^{-5}$	2142

shooting cycle, it takes less GMRES iterations per Newton step and less CPU time, as the first four shooting cycles need as much time as the first cycle without refinement.

Furthermore, finer spatial meshes lead to a slight increase of GMRES iterations if the global refinement strategy is applied. These observations hold also for the case where the adaptive thinning method is included into the shooting process. The corresponding results are displayed in Table 7.20, which also contains information on the number of shooting intervals in the different shooting cycles.

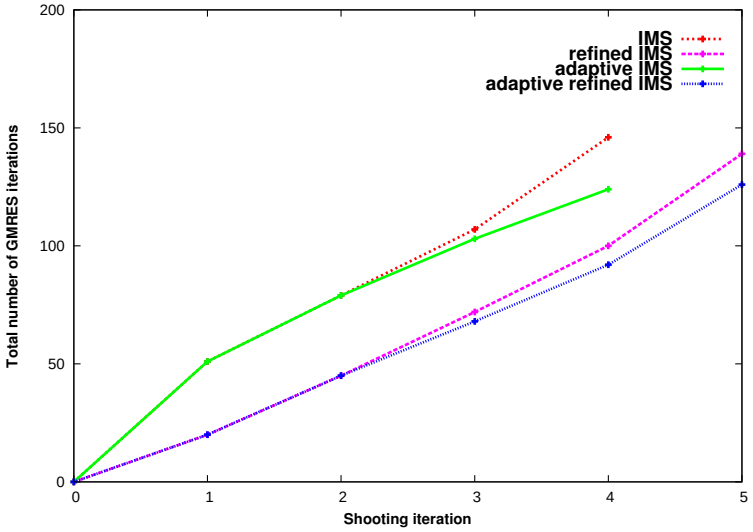

Figure 7.12. Example 7.9: Number of GMRES iterations for four different solution approaches: without refinement nor adaptivity, with adaptivity only, with refinement only, and with both refinement and adaptivity.

Figures 7.12 and 7.13 illustrate the most important results from the above tables. The total number of GMRES iterations required during the shooting process is depicted in Figure 7.12. As the adaptive approaches start on the same equidistant shooting grid, the curves for IMS and adaptive IMS respectively for refined IMS and adaptive refined IMS coincide for the first shooting iterations. When the thinning mechanism takes effect, the curves begin to differ and the adaptive approaches require less GMRES iterations due to the diminished shooting system.
In Figure 7.13 the CPU times taken by the different IMS variants are compared. As a result, the adaptive IMS approach converges faster than the unmodified IMS method, but adaptivity by itself is less efficient than the global mesh refinement strategy. However, the combination of global refinement and adaptivity displays the fastest convergence while providing the same accuracy with respect to both objective functional values and shooting residuals.

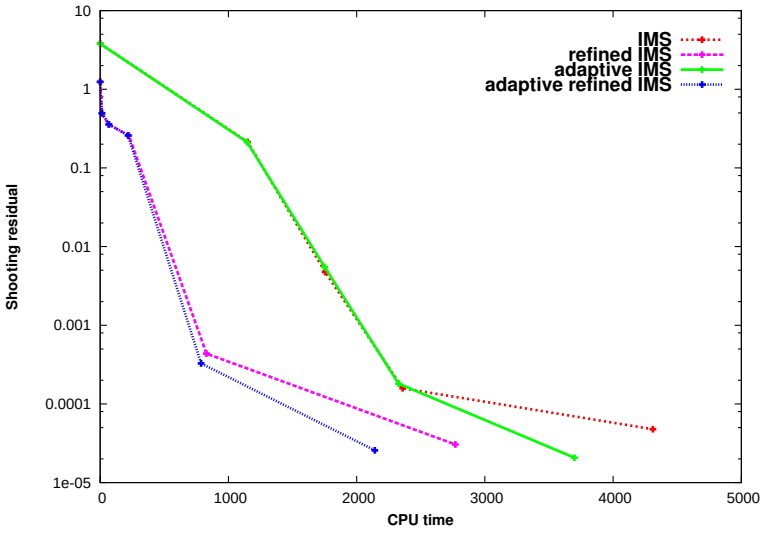

Figure 7.13. Example 7.9: CPU time for four different solution approaches: without refinement nor adaptivity, with adaptivity only, with refinement only, and with both refinement and adaptivity.

The second test problem revisits Example 6.1 from Section 6.3. The main differences in comparison to Example 7.9 are the distributed tracking type functional and the presence of additional control box constraints.

Example 7.10. *Consider the problem*

$$\min_{(q,u)} \; J(q,u) = \frac{1}{2}\int_0^T \|u(x,t) - \hat{u}(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t + \frac{\alpha}{2}\int_0^T \|q(x,t)\|^2_{L^2(\Omega)}\,\mathrm{d}t,$$

subject to the nonstationary nonlinear Helmholtz problem

$$\begin{aligned} \partial_t u(x,t) - \Delta u(x,t) - \omega u(x,t) + u^3(x,t) &= q(x,t) && \text{in } \Omega \times (0,T], \\ u(x,t) &= 0 && \text{on } \partial\Omega \times [0,T], \\ u(x,0) &= u_0(x) && \text{in } \Omega \end{aligned}$$

and the constant box constraints

$$-0.5 \le q(x,t) \le 0.5 \;\; \text{a. e. in } \Omega \times [0,T].$$

In this example, the heuristic selection parameter for removing shooting points from the grid is fixed at $\alpha_{\text{low}} = \frac{1}{10}$. As before, four different indirect shooting variants are compared. The indirect shooting method is used without any additional features, then it is complemented by global space mesh refinement, by the adaptive thinning strategy, and finally by a combination of both. In contrast to Example 7.9, if spatial mesh refinement is used we perform two shooting cycles on each refinement level.

Table 7.21. Example 7.10: IMS on 10 equidistant shooting intervals without (left) and with global refinement (right) as proposed in Subsection 5.5.3; number of GMRES iterations, development of functional values and shooting residual, CPU time (criterion: $\|F\|_2 < 10^{-3}$).

	without refinement				with refinement				
iter.	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)	ref.	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)
0	–	2.391	$3.9 \cdot 10^{0}$	–	1	–	1.582	$9.8 \cdot 10^{-1}$	–
1	22	1.618	$2.4 \cdot 10^{-1}$	901	2	19	1.508	$4.9 \cdot 10^{-1}$	29
2	21	1.758	$7.6 \cdot 10^{-2}$	1773	2	21	1.693	$2.1 \cdot 10^{-1}$	106
3	24	1.801	$2.7 \cdot 10^{-2}$	3670	3	21	1.790	$5.8 \cdot 10^{-2}$	325
4	24	1.817	$9.5 \cdot 10^{-3}$	5594	3	24	1.817	$1.1 \cdot 10^{-2}$	2230
5	24	1.823	$3.5 \cdot 10^{-3}$	7521	4	24	1.824	$2.9 \cdot 10^{-3}$	4135
6	24	1.825	$1.7 \cdot 10^{-3}$	9446	4	24	1.825	$1.1 \cdot 10^{-3}$	6016
7	24	1.826	$4.6 \cdot 10^{-4}$	11373	4	24	1.826	$3.8 \cdot 10^{-4}$	7923

As in Example 7.9, the results are summarized in two tables comparing equidistant IMS with and without spatial mesh refinement as well as residual-based adaptive IMS, also with and without employing the global mesh refinement strategy. Table 7.21 presents the results for equidistant shooting on 10 intervals. Note that, due to Table 6.2, at least six intervals are required for solving the problem. The table confirms that the employment of the spatial mesh refinement increases numerical efficiency. In the first shooting cycles, slightly less GMRES iterations are performed than in the IMS method without refinement, and there is a saving of CPU time of one third.
Table 7.22 displays the corresponding results for IMS with the adaptive thinning strategy. In contrast to the previous example, the adaptive approach requires more numerical effort in terms of CPU time than the original IMS method, even though the shooting grid is reduced to the optimal number of six intervals that are, however, not equidistantly distributed. The shooting grid is reduced from the initial

$$[0;\ 0.5;\ 1;\ 1.5;\ 2;\ 2.5;\ 3;\ 3.5;\ 4;\ 4.5;\ 5]$$

to the grid

$$[0;\ 1;\ 1.5;\ 2.5;\ 3;\ 4;\ 5]$$

in the third shooting cycle. Presumably, this new shooting point distribution worsens the instability of the shooting algorithm. Thus, the solution of the linearized BVP (5.11) and (5.12) takes longer than on an equidistant shooting grid of six intervals. This constitutes a basic problem of the thinning mechanism, as it does not take the conditioning or stability of the underlying problem into account, although it reduces the number of shooting intervals and leads to a significant decrease in GMRES iterations.

Table 7.22. Example 7.10: Residual-based adaptive IMS without (left) and with global refinement (right) as proposed in Subsection 5.5.3; the refinement levels are the same as in Table 7.21. The number of shooting intervals (SI) is also displayed (criterion: $\|F\|_2 < 10^{-3}$).

	without refinement					with refinement				
iter.	$\#_{\text{SI}}$	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)	$\#_{\text{SI}}$	$\#_{\text{it}}$	$J(q,u)$	$\|F\|_2$	t(s)
0	10	–	2.391	$3.9 \cdot 10^{0}$	–	10	–	1.582	$9.8 \cdot 10^{-1}$	–
1	10	21	1.618	$2.4 \cdot 10^{-1}$	865	10	19	1.508	$4.9 \cdot 10^{-1}$	28
2	10	21	1.750	$8.3 \cdot 10^{-2}$	1713	10	20	1.440	$1.6 \cdot 10^{-1}$	103
3	6	13	1.796	$3.2 \cdot 10^{-2}$	4149	9	18	1.756	$1.1 \cdot 10^{-1}$	183
4	6	13	1.815	$1.3 \cdot 10^{-2}$	6522	9	19	1.748	$4.0 \cdot 10^{-2}$	446
5	6	13	1.822	$5.2 \cdot 10^{-3}$	8872	8	17	1.824	$1.4 \cdot 10^{-2}$	1025
6	6	14	1.825	$2.0 \cdot 10^{-3}$	11292	8	17	1.825	$3.1 \cdot 10^{-3}$	4247
7	6	14	1.826	$7.4 \cdot 10^{-4}$	13708	8	17	1.829	$8.8 \cdot 10^{-4}$	7139

The combination of the thinning mechanism with global space mesh refinement displayed in the right panel of Table 7.22 provides better results with respect to CPU time. The number of shooting intervals does not decrease largely during the process, but the smaller shooting system entails a noticeable reduction of the number of GMRES iterations. The combined method takes slightly less CPU time until the tolerance of 10^{-3} is reached as compared to the pure spatial mesh refinement strategy.

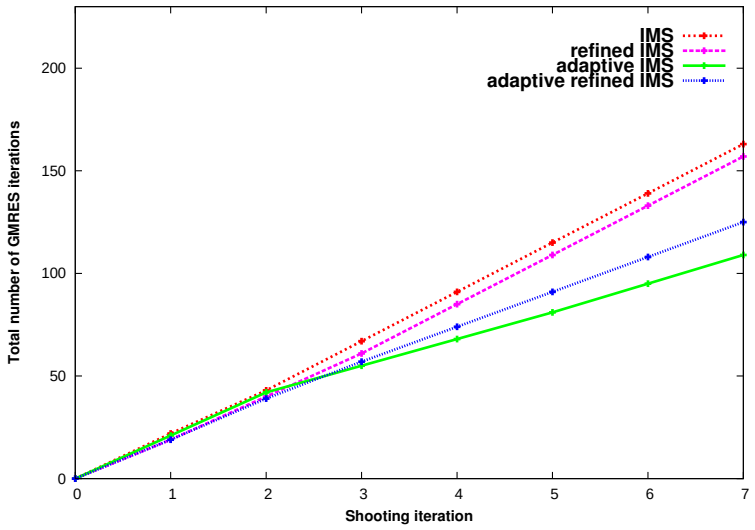

Figure 7.14. Example 7.10: Number of GMRES iterations for four different solution approaches: without refinement nor adaptivity, with adaptivity only, with refinement only, and with both refinement and adaptivity.

As in Example 7.9, both the total number of GMRES iterations taken by the different IMS variant and the corresponding CPU times are illustrated in Figures 7.14 and 7.15, respectively.

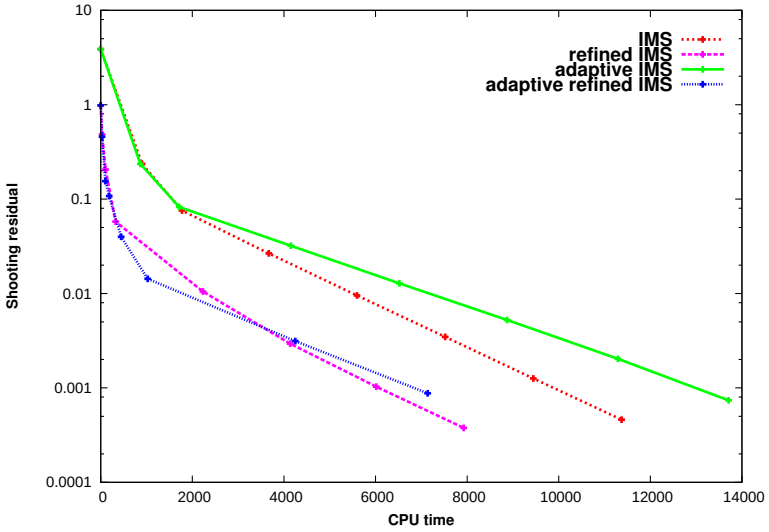

Figure 7.15. Example 7.10: CPU time for four different solution approaches: without refinement nor adaptivity, with adaptivity only, with refinement only, and with both refinement and adaptivity.

Finally, we give a brief summary of this section. In Example 7.9 where no additional constraints were imposed on the control, we observed that the thinning approach led to a decrease of numerical effort in terms of Newton-GMRES iterations for the shooting system as well as to a decrease in CPU time. This effect is more pronounced if the adaptive shooting technique is combined with the global space mesh refinement proposed in Subsection 5.5.3.

Remark 7.6. The solution of the considered example displays no special structure in the computational domain $\Omega \times I$ but is spatially and temporally distributed. For solutions with a special structure such as a rotating bump with a small support as in Example 2.4, the global refinement strategy might be unable to resolve the solution adequately. In this case, local mesh refinement, e. g., based on a dual weighted residual (DWR) approach as proposed by Becker & Rannacher [5] is advisable.
Furthermore, in the example only one Newton-GMRES iteration is carried out on each mesh refinement level. A procedure capable of balancing the shooting residual (and by this the shooting grid adaptation) and the space mesh refinement might further improve the results.

In the second Example 7.10 additional constant box constraints were prescribed. The residual-based adaptive shooting approach led to a significant decrease of the number of

Newton-GMRES iterations in later shooting cycles where the shooting grid was reduced. However, the adaptive method proved slightly less efficient than equidistant multiple shooting with respect to CPU time. The global space mesh refinement provided an increase in computational efficiency, as was already observed for all nonlinear examples in Chapters 5 and 6. The combination of adaptivity and spatial mesh refinement provided a similar result as the global refinement strategy alone.
In summary, the examples of this section suggest that there is a computational benefit in using the adaptive thinning strategy for unconstrained parabolic OCP. A combination of global space mesh refinement and the adaptive thinning approach is computationally most efficient. In the control constrained case, this benefit has not been observed. However, in all cases the use of refinement strategies for the spatial mesh increases the computational efficiency both with respect to the number of Newton-GMRES iterations and with respect to CPU time.

8 Conclusion and Outlook

Conclusion. This thesis provides a thorough examination of existing multiple shooting approaches for parabolic OCP as well as extensions of the method in several directions. Both indirect and direct shooting variants are formulated within a common abstract framework, enabling their efficient application to complex problems. We adapt different modern techniques for coping with control constraints to the multiple shooting context. Results for nonlinear constrained control problems indicate a superiority of direct shooting methods. The development of two different adaptive shooting techniques constitutes a novelty even for ODE problems. In the PDE case, we combine the residual-based adaptive approach with a global mesh refinement strategy, which improves the computational efficiency of the indirect shooting method significantly.

Different shooting methods. The first main issue of this work, providing the basis for all further developments, was to enhance several existing multiple shooting approaches for solving BVP and OCP governed by parabolic PDE and to elaborate their interdependencies. By extending the abstract control problem on the function space level, we provided a common framework for both direct and indirect shooting methods. These approaches are based on different splittings of the first order optimality system of the extended OCP.
Furthermore, we demonstrated that a classical DMS variant which is common in ODE optimal control constitutes a reduced formulation of our DMS approach. These results are contained in Chapter 5 and provide a detailed survey of different multiple shooting approaches. Our abstract presentation is complemented by concrete algorithmic realizations of the IMS and DMS methods. These algorithms could be further enhanced, e. g., by suitable preconditioners for the shooting system, by condensing techniques, or by employing different discretizations for the state and control variables as proposed in Chapter 4.

Control constrained problems. Both IMS and DMS methods were applied to parabolic OCP with additional constraints on the control variable in Chapter 6. Therefore, we reformulated the OCP in a way that makes it accessible to multiple shooting techniques. We employed projection methods to cope with the constraints, but also adapted modern primal-dual active set strategies to the shooting framework.
Our numerical results for both unconstrained and control constrained OCP showed that in most cases DMS is more efficient than IMS, particularly for nonlinear constrained problems. Our experience confirms the preference of direct over indirect shooting methods that can be observed in ODE optimal control. As global constraints, e. g., on the control mean-value, are usually not localizable, they are difficult to include into the multiple shooting context. Therefore, they were not treated in this work.

Adaptive shooting. Numerical results for both ODE and PDE test cases led to the conjecture that shooting grids can be optimized by choosing the number and position of shooting points appropriately. Of course, a large number of equidistant shooting intervals usually leads to a stable algorithm; however, with an increasing number of shooting points the numerical effort grows significantly in the PDE framework. To avoid costly trial and error computations, adaptive techniques for determining the shooting grid are necessary, but they have to take the mentioned trade-off into account.
There are only few results on adaptive shooting methods; therefore, we focussed on developing two different approaches in Chapter 7. The first one is based on bounding the sensitivities of the problem and involves no a priori information on the shooting grid. The second one is residual-based and starts from a fine shooting grid which is then successively thinned out. We tested both methods for ODE boundary value and control problems. As the residual-based approach proved more efficient and has both theoretical and implementational advantages over the sensitivity based one, we decided to skip the latter in the PDE framework.
The residual-based adaptive method was transferred to PDE control problems, including additional control constraints. As in the PDE context, fine shooting grids are prohibitive due to their large computational costs, the approach is combined with global mesh refinement. The computation starts with many shooting intervals but on a coarse space mesh, and as the number of shooting intervals is decreased, the space mesh is refined. Several research opportunities concerning further adaptive features in the multiple shooting framework are addressed in the corresponding part of the outlook below.

Outlook. Finally, we provide an outlook to subjects that are related to our work and raise interesting questions that could enhance future research. We motivate them by connecting them to the topics treated in this thesis.

Parallelization. An important feature of multiple shooting methods that we covered in several remarks throughout the thesis is their potential for time parallel computing. Although there are some earlier publications on the subject (cf. Nievergelt [89], Kramer & Mattheij [71] or Kiehl [67]), it has gained much attention only after Lions et al. [76] proposed the so-called 'parareal' method. The latter has been employed for ODE problems by Guibert & Tromeur-Dervout [48], for PDE governed OCP by Maday & Turinici [79], and for several application problems by Bal and co-authors [3, 4]. Gander & Vandewalle [41] established the connection to multiple shooting. In recent years, the parareal method was also used as a preconditioner by Ulbrich [110, 111]. A historical survey of parallel time domain decomposition methods is provided by Gander [42].
Domain decomposition methods for parallel computing in the spatial variables have been thoroughly examined (see, e. g., Toselli & Widlund [107]), but complex nonstationary applications such as three-dimensional models in biology or medicine also require parallelization in time. Multiple shooting is a promising technique in this regard.
Therefore, the parallelizability of the features developed in this thesis was one of our concerns. This contributed to our rejection of the sensitivity based adaptive approach in Chapter 7. Instead we focussed on the residual-based technique in the PDE framework as it can be parallelized.

State constraints. Constraints on the state variable or the gradient of the state constitute another possible extension of our research. For elliptic OCP, Schiela & Wollner [102] proposed barrier methods to cope with constrained state gradients. A modification of their approach for parabolic OCP could be adapted to the needs of multiple shooting, thereby extending our shooting algorithms from Chapter 6.

Further adaptive features. We combined our residual-based adaptive shooting algorithm with global space mesh refinement in Chapter 7 in order to improve its computational efficiency. To better resolve structural features of a given problem, one might use local mesh adaptation techniques instead. Hesse & Kanschat [53] proposed a dual weighted residual (DWR) based adaptive approach on equidistant shooting grids, where additional projection errors occur at the shooting points. However, their algorithm adapts the space mesh only after the multiple shooting process is converged; local mesh refinement in each shooting cycle might improve the efficiency of the method.
A second application of DWR strategies in the multiple shooting framework relies on results by Meidner et al. [86] and Rannacher & Vihharev [95]. Extending previous work of Becker et al. [6] on iterative multigrid solvers, they showed a way to balance discretization and iteration errors for linear and nonlinear iterative solution methods via DWR error estimators. Their algorithms provide efficient stopping criteria for iterative solvers which avoid unnecessary iterations when the discretization error becomes dominant. As the multiple shooting algorithms from Chapter 5 include several iterative processes (e. g., the Newton-CG solver for subinterval OCP or the Newton-GMRES solver for the shooting system), efficient stopping criteria are desirable to avoid computational overhead. However, the multiple shooting method is not of Galerkin type, which is crucial for employing DWR techniques. This is a challenging topic for future research.

Applications. Finally, the numerical examples considered in this thesis are test cases for our theoretical developments. We treated both linear and nonlinear problems, thereby including control constraints and performing adaptive shooting techniques. An important topic for future research is the employment of the developed methods in real-world applications. In this regard, recent articles by Richter & Wick [98], Hasegawa [49] or Klinger [68] open up new perspectives of how to apply multiple shooting to fluid structure interaction problems, turbulent flow problems, and image processing problems, respectively. Both parallelization and adaptivity are desirable features for complex applications, e. g., three-dimensional computations on highly resolved meshes.

Bibliography

[1] J. Albersmeyer. *Adjoint-based algorithms and numerical methods for sensitivity generation and optimization of large scale dynamic systems*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2010.

[2] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, volume 13 of *Classics in Applied Mathematics*. SIAM, 1995.

[3] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Comp. Science and Engrg.*, pages 425–432. Springer, 2005.

[4] G. Bal and Y. Maday. A 'parareal' time discretization for nonlinear PDEs with application to the pricing of an American put. In *Recent Developments in Domain Decomposition Methods*, volume 23 of *Lecture Notes in Comp. Science and Engrg.*, pages 189–202. Springer, 2002.

[5] 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.

[6] R. Becker, C. Johnson, and R. Rannacher. Adaptive error control for multigrid finite element methods. *Computing*, 55(4):271–288, 1995.

[7] R. Becker, D. Meidner, and B. Vexler. Efficient numerical solution of parabolic optimization problems by finite element methods. *Optim. Method. Softw.*, 22(5): 813–833, 2007.

[8] D. Beigel. *Efficient goal-oriented global error estimation for BDF-type methods using discrete adjoints*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2012.

[9] M. Bergounioux, K. Ito, and K. Kunisch. Primal-dual strategy for constrained optimal control problems. *SIAM J. Control Optim.*, 37(4):1176–1194, 1999.

[10] H. G. Bock. Zur numerischen Behandlung zustandsbeschränkter Steuerungsprobleme mit Mehrzielmethode und Homotopieverfahren. *Zeitschrift für Angewandte Mathematik und Mechanik*, 57(4):266–268, 1977.

[11] H. G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series in Chemical Physics*, pages 102–125. Springer, 1981.

[12] H. G. Bock. *Optimierungsverfahren - Software und praktische Anwendungen*. Carl-Crantz-Gesellschaft, Oberpfaffenhofen, 1981.

[13] H. G. Bock. Recent advances in parameter identification problems for ODE. In *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pages 95–121. Birkhäuser, 1983.

[14] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften*. Universität Bonn, 1987.

[15] H. G. Bock and K. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress Budapest*. Pergamon Press, 1984.

[16] H. G. Bock, E. Kostina, and J. Schlöder. Direct multiple shooting for optimization problems in ODE models. In *Multiple Shooting and Time Domain Decomposition Methods*. Springer, 2015 (to appear).

[17] D. Braess. *Finite Elemente*. Springer, 4th edition, 2007.

[18] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, 2nd edition, 2002.

[19] R. Bulirsch. *Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung*. Carl-Crantz-Gesellschaft, Heidelberg, 1973.

[20] R. Bulirsch and J. Stoer. *Introduction to Numerical Analysis*, volume 12 of *Texts in Applied Mathematics*. Springer, 3rd edition, 2002.

[21] T. Carraro and M. Geiger. Direct and indirect multiple shooting for parabolic optimal control problems. In *Multiple Shooting and Time Domain Decomposition Methods*. Springer, 2015 (to appear).

[22] T. Carraro, M. Geiger, and R. Rannacher. Indirect multiple shooting for nonlinear parabolic optimal control problems with control constraints. *SIAM J. Sci. Comput.*, 36(2):A452–A481, 2014.

[23] Y. Choi, C. Farhat, W. Murray, and M. Saunders. A practical factorization of a Schur complement for PDE-constrained distributed optimal control. *Journal of Scientific Computing (published online; DOI: 10.1007/s10915-014-9976-0)*, 2014.

[24] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 40 of *Classics in Applied Mathematics*. SIAM, 2002.

[25] E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

[26] A. Comas. *Time-Domain Decomposition Preconditioners for the Solution of Discretized Parabolic Optimal Control Problems*. PhD thesis, Rice University, 2005.

[27] W. A. Coppel. Dichotomies and reducibility. *J. of Diff. Eq.*, 3(4):500–521, 1967.

[28] W. A. Coppel. *Dichotomies in stability theory*, volume 629 of *Lecture Notes in Mathematics*. Springer, 1978.

[29] B. Dacorogna. *Direct Methods in the Calculus of Variations*, volume 78 of *Applied Mathematical Sciences*. Springer, 2nd edition, 2008.

[30] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology vol. 5: Evolution Problems*, volume 5. Springer, 1992.

[31] F. de Hoog and R. M. M. Mattheij. On dichotomy and well conditioning in BVP. *SIAM J. Numer. Anal.*, 24(1):89–105, 1987.

[32] P. Deuflhard. A modified newton method for the solution of ill-conditioned systems of nonlinear equations with applications to multiple shooting. *Numer. Math.*, 22(4): 289–311, 1974.

[33] P. Deuflhard. A stepsize control for continuation methods and its special application to multiple shooting techniques. *Numer. Math.*, 33(2):115–146, 1979.

[34] P. Deuflhard. *Newton Methods for Nonlinear Problems*, volume 35 of *Springer Series in Computational Mathematics*. Springer, 2004.

[35] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast Motions in Biomechanics and Robotics*, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 65–93. 2006.

[36] J. C. Dunn. Global and asymptotic convergence rate estimates for a class of projected gradient processes. *SIAM J. Control Optim.*, 19(3):368–400, 1981.

[37] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers, 2000.

[38] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Cambridge University Press, 1996.

[39] L. C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 1998.

[40] A. V. Fursikov. *Optimal Control of Distributed Systems. Theory and Applications*, volume 187 of *Translations of Mathematical Monographs*. American Mathematical Society, 2000.

[41] M. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.

[42] M. J. Gander. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods*. Springer, 2015 (to appear).

[43] C. Geiger and Ch. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben.* Springer, 2002.

[44] J. H. George and R. W. Gunderson. Conditioning of linear boundary value problems. *BIT*, 12(2):172–181, 1972.

[45] T. R. Goodman and G. N. Lance. The numerical integration of two-point boundary value problems. *Math. Tables Other Aids Comp.*, 10(54):82–86, 1956.

[46] R. Griesse and B. Vexler. Numerical sensitivity analysis for the quantity of interest in PDE-constrained optimization. *SIAM J. Sci. Comput.*, 29(1):22–48, 2007.

[47] R. Griesse and S. Volkwein. A primal-dual active set strategy for optimal boundary control of a nonlinear reaction-diffusion system. *SIAM J. Control Optim.*, 44(2): 467–494, 2005.

[48] D. Guibert and D. Tromeur-Dervout. Adaptive parareal for systems of ODEs. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lecture Notes in Comp. Science and Engrg.*, pages 587–594. Springer, 2007.

[49] Y. Hasegawa. Optimal control of heat and fluid flow for efficient energy utilization. In *Multiple Shooting and Time Domain Decomposition Methods.* Springer, 2015 (to appear).

[50] M. Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *J. Comput. Appl. Math.*, 173(1):169–198, 2005.

[51] R. Herzog and K. Kunisch. Algorithms for PDE-constrained optimization. *GAMM Reports*, 33(2):163–176, 2010.

[52] H. K. Hesse. *Multiple Shooting and Mesh Adaptation for PDE Constrained Optimization Problems.* PhD thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2008.

[53] H. K. Hesse and G. Kanschat. Mesh adaptive multiple shooting for partial differential equations. Part I: linear quadratic optimal control problems. *J. Numer. Math.*, 17 (3):195–217, 2009.

[54] H. K. Hesse and R. Rannacher. Direct vs. indirect multiple shooting for nonstationary PDE-based optimization problems. *Preprint*, 2008.

[55] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49(6):409–436, 1952.

[56] N. T. Hieu. Remarks on the shooting method for nonlinear two-point boundary value problems. *VNU Journal of Science*, 3:18–25, 2003.

[57] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM J. Optim.*, 13(3):865–888, 2002.

[58] M. Hinze. A variational discretization concept in control constrained optimization: The linear-quadratic case. *Comput. Optim. Appl.*, 30(1):45–61, 2005.

[59] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, 2009.

[60] K. Ito and K. Kunisch. *Lagrange Multiplier Approach to Variational Problems and Applications*, volume 15 of *Advances in Design and Control*. SIAM, 2008.

[61] T. Jankowski. Approximate solutions of boundary value problems for systems of ordinary differential equations. *Zh. Vychisl. Mat. Mat. Fiz.*, 35(7):1050–1057, 1995.

[62] Ch. Kanzow. *Numerik linearer Gleichungssysteme: Direkte und iterative Verfahren*. Springer, 2005.

[63] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinit linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1300–1317, 2000.

[64] H. B. Keller. *Numerical Solution of Two-Point Boundary Value Problems*, volume 24 of *CBMS Regional Conference Series in Applied Mathematics*. SIAM, 1976.

[65] C. T. Kelley. *Iterative Methods for Optimization*, volume 18 of *Frontiers in Applied Mathematics*. SIAM, 1987.

[66] C. T. Kelley and E. W. Sachs. Solution of optimal control problems by a pointwise projected newton method. *SIAM J. Control Optim.*, 33(6):1731–1757, 1995.

[67] M. Kiehl. Parallel multiple shooting for the solution of initial value problems. *Parallel Computing*, 20(3):275–295, 1994.

[68] M. Klinger. A variational approach for physically based image interpolation across boundaries. In *Multiple Shooting and Time Domain Decomposition Methods*. Springer, 2015 (to appear).

[69] S. Körkel, E. Kostina, H. G. Bock, and J. P. Schlöder. Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes. *Optim. Method. Softw.*, 19(3):327–338, 2004.

[70] M. E. Kramer. *Aspects of solving non-linear boundary value problems numerically*. PhD thesis, Technische Universiteit Eindhoven, 1992.

[71] M. E. Kramer and R. M. M. Mattheij. Application of global methods in parallel shooting. *SIAM J. Numer. Anal.*, 30(6):1723–1739, 1993.

[72] K. Kunisch and A. Rösch. Primal-dual active set strategy for a general class of constrained optimal control problems. *SIAM J. Optim.*, 13(2):321–334, 2002.

[73] D. B. Leineweber. The Theory of MUSCOD in a Nutshell. Master's thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 1995.

[74] D. B. Leineweber, I. Bauer, H. G. Bock, and J. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: Theoretical aspects. *Comp. Chem. Eng.*, 27(2):157–166, 2003.

[75] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*, volume 170 of *Grundlehren Math. Wiss.* Springer, 1971.

[76] J.-L. Lions, Y. Maday, and G. Turinici. Resolution d'edp par un schéma en temps 'pararéel'. *C. R. Acad. Sci. Paris Ser. I 332*, 335(4):661–668, 2001.

[77] P. Lory. Enlarging the domain of convergence for multiple shooting by the homotopy method. *Numer. Math.*, 35(2):231–240, 1980.

[78] D. G. Luenberger. *Optimization by Vector Space Methods*. Series in Decision and Control. Wiley Interscience, 1969.

[79] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C. R. Acad. Sci. Paris Ser.*, 335(4):387–392, 2002.

[80] M. R. Maier. An adaptive shooting method for singularly perturbed boundary value problems. *SIAM J. Sci. Stat. Comput.*, 7(2):418–440, 1986.

[81] R. M. M. Mattheij. Estimates for the errors in the solutions of linear boundary value problems due to perturbations. *Computing*, 27(4):299–318, 1981.

[82] R. M. M. Mattheij. The conditioning of linear boundary value problems. *SIAM J. Numer. Anal.*, 19(5):963–978, 1982.

[83] R. M. M. Mattheij and G. W. M. Staarink. On optimal shooting intervals. *Math. Comput.*, 42(165):25–40, 1984.

[84] D. Meidner. *Adaptive space-time finite element methods for optimization problems governed by nonlinear parabolic systems*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2008.

[85] D. Meidner and B. Vexler. Adaptive space-time finite element methods for parabolic optimization problems. *SIAM J. Control Optim.*, 46(1):116–142, 2007.

[86] D. Meidner, R. Rannacher, and J. Vihharev. Goal-oriented error control of the iterative solution of finite element equations. *J. Numer. Math.*, 17(2):143–172, 2009.

[87] A. Meister. *Numerik linearer Gleichungssysteme*. Vieweg, 3rd edition, 2008.

[88] D. D. Morrison, J. D. Riley, and J. F. Zancanaro. Multiple shooting method for two-point boundary value problems. *Communications of the ACM*, 5(12):613–614, 1962.

[89] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12):731–733, 1964.

[90] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operation Research and Financial Engineering. Springer, 2nd edition, 2006.

[91] F. W. J. Olver. *Asymptotics and Special Functions*. Academic Press, 1974.

[92] M. R. Osborne. On shooting methods for boundary value problems. *J. Math. Anal. Appl.*, 27(2):417–433, 1969.

[93] A. Potschka. Handling Path Constraints in a Direct Multiple Shooting Method for Optimal Control Problems. Master's thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2006.

[94] A. Potschka. *A direct method for the numerical solution of optimization problems with time-periodic PDE constraints*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2012.

[95] R. Rannacher and J. Vihharev. Adaptive finite element analysis of nonlinear problems: balancing of discretization and iteration errors. *J. Numer. Math.*, 21(1):23–62, 2012.

[96] A. V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[97] A. V. Rao and K. D. Mease. Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems. *Optim. Control Appl. Meth.*, 21(1): 1–19, 2000.

[98] T. Richter and T. Wick. On time discretizations of fluid-structure interactions. In *Multiple Shooting and Time Domain Decomposition Methods*. Springer, 2015 (to appear).

[99] P. E. Rutquist and M. M. Edvall. *PROPT - Matlab Optimal Control Software. TOMLAB Optimization*, 2010. URL `http://tomopt.com/docs/TOMLAB_PROPT.pdf`.

[100] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, second edition, 2003.

[101] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[102] A. Schiela and W. Wollner. Barrier methods for optimal control problems with convex nonlinear gradient state constraints. *SIAM J. Optim.*, 21(1):269–286, 2011.

[103] M. Schmich. *Adaptive finite element methods for computing nonstationary incompressible flows*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Fakultät für Mathematik und Informatik, 2009.

[104] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.*, 30(1):369–393, 2008.

[105] R. Serban, S. Li, and L. Petzold. Adaptive algorithms for optimal control of time-dependent partial differential-algebraic systems. *Int. J. Numer. Meth. Eng.*, 57(10): 1457–1469, 2003.

[106] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20(3):626–637, 1983.

[107] A. Toselli and O. B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.

[108] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, 2010.

[109] M. Ulbrich and S. Ulbrich. *Nichtlineare Optimierung*. Mathematik Kompakt. Birkhäuser, 2012.

[110] S. Ulbrich. Generalized SQP-methods with 'parareal' time-domain decomposition for time-dependent PDE-constrained optimization. In *Real-time PDE-constrained optimization*, volume 3 of *Computational Science and Engineering*, pages 145–168. SIAM, 2007.

[111] S. Ulbrich. Preconditioners based on 'parareal' time-domain decomposition for time-dependent PDE-constrained optimization. In *Multiple Shooting and Time Domain Decomposition Methods*. Springer, 2015 (to appear).

[112] B. Vexler and W. Wollner. Adaptive finite elements for elliptic optimization problems with control constraints. *SIAM J. Control Optim.*, 47(1):509–534, 2008.

[113] L. N. Vicente. Derivative computations for a class of optimal control problems. Technical report, School of Finite Elements and Applications (Centro Internacional de Matematica), Coimbra, 1998.

[114] R. Weiss. The convergence of shooting methods. *BIT*, 13(4):470–475, 1973.

[115] J. Wloka. *Partial Differential Equations*. Cambridge University Press, 1987.

Acknowledgments

A thesis like the one at hand may be authored by one person, but there are always many people in the background contributing in different ways to its genesis. It is my pleasure to finally share some heartfelt thank-yous.

In first place, I am deeply grateful to my supervisor, Prof. Dr. Dr. h. c. Rolf Rannacher, who not only proposed the subject of this work, but also supported me in many ways during the last years. I am especially obliged to him for giving me the opportunity to co-organize a workshop on 'Multiple Shooting and Time Domain Decomposition Methods' (held in Heidelberg from May 6^{th} to 8^{th}, 2013), and to afterwards co-publish a Springer theme issue of the same title which is now due to appear. Furthermore, under his guidance I was able to gain experience in writing articles, giving talks and doing all things that are important in research.
The Numerical Analysis group in Heidelberg always provided a most pleasant working atmosphere, for which I am grateful to all my collegues past and present. In particular, I thank Christian Goll, Daniel Gerecht, Felix Brinkmann, Ina Schüssler, Matthias Maier, Sara Lee, Stefan Frei and Sven Wetterauer who share the same fate with me as PhD students, and Dr. Adrian Hirn, Dr. Dominik Meidner, Dr. Elfriede Friedmann, Dr. Jevgeni Vihharev and Prof. Dr. Thomas Richter who already advanced successfully and set a good example. All of them were always open for discussion and often for nonsense, which delivered a fruitful background for work and life beyond.
Two collegues whom I am happy to count among my friends stand out and shall be mentioned separately. One of them, Matthias Klinger, has been my office mate over the last years who by means of his humor and cheerfulness grounded me more than once when I felt that I began to stagger. He patiently listened to both technical and private problems and helped me a lot, also by reading and commenting talks, papers and this thesis. The second one is Dr. Thomas Carraro, with whom I had the good fortune to work closely together. He taught me much of what I know about optimization and adaptivity, he never grew tired discussing the most obvious things until I really understood them, he shared his professional experience in workshop organization, compact course preparation, book editing, and he simply is the most genuinely interested researcher I met during my PhD time. Still more important, he is a deeply humane person and, in every respect, a role model.
I would also like to express my gratitude to my second supervisor, Prof. Dr. Dr. h. c. Hans Georg Bock, his former PhD students Dr. Dörte Beigel and Dr. Andreas Potschka, as well as Dr. Stefan Körkel, who all helped me understand the direct multiple shooting method. I am further indebted to Prof. Dr. Hans Josef Pesch from Bayreuth, who invited me for a talk and afterwards encouraged me to pursue my studies on adaptive multiple shooting.

In the first half of my PhD time, financial support came through the former SPP 1253 ('Optimization with Partial Differential Equations') by the German Research Foundation (DFG), and later on the Heidelberg Graduate School (HGS) provided travel funds. I owe a great deal of the opportunities I had in the past years to these two organizations.

Finally, there are those people most important in my life, my friends, partner, and family, who always gave me invaluable support and whom I would like to thank in German:
Ich danke den Aschaffenburgern, insbesondere Gabriel für die wesentliche Freundschaft in meinem Leben, sowie Peggy, Melissa, Tobi, Flo und Thorsten für die Spieleabende, Gespräche und Spaziergänge, die für mich unverzichtbar waren.
Weiterhin bin ich den Heidelbergern dankbar, darunter vor allem Saskia, Cilly, Cathy sowie Timo und Ulli, die mich in allen Lebenslagen kennengelernt und nie allein gelassen haben.
Sehr wichtig ist es mir, Gabi Schwarz für mehr als fünf Jahre Partnerschaft und Freundschaft zu danken, die mir so viel Ausgeglichenheit und Stabilität verliehen haben und ohne die ich diese Arbeit nicht zu Ende gebracht hätte.
Zu meinem Bruder Thomas habe ich nicht nur ein gutes geschwisterliches Verhältnis, er ist mir auch einer der besten Freunde und hat immer ein offenes Ohr, in privaten wie in fachlichen Fragen. Dafür von Herzen vielen Dank!
Ich habe das große Glück, in meiner Partnerin Sara zugleich eine der besten Freundinnen gefunden zu haben, die ich mir vorstellen kann! Die Geduld und bedingungslose Unterstützung, die ich durch sie erfahre, kann ich mit Worten nicht genug wertschätzen. Ich kann nur einfach 'Danke' sagen.
Am Ende der langen Liste stehen drei Menschen, denen ich zu verdanken habe, was ich heute bin. Einmal sind das meine Eltern, Gertrud und Peter Geiger, die mir immer geholfen und mich niemals im Stich gelassen haben, die mir gute Zeiten geschenkt und schwierige klaglos ertragen haben. Wie sollte ich ausdrücken, was ich für sie empfinde? Zum Anderen ist es meine Großmutter, Kunigunde Schuck, die mir so viel bedeutet und der ich sagen möchte: Oma, die wichtigen Dinge im Leben habe ich von Dir gelernt! Danke!

"*It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way – in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.*"

(from: Charles Dickens, A Tale of Two Cities)