Efficient direct discretization strategies for re-entry optimal control problems with minimum heating

Blanca Pablos Martín

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Gutachter: 1. Prof. Dr. rer. nat. Matthias Gerdts
2. Prof. Dr. rer. nat. Kurt Chudej

Die Dissertation wurde am 8. Juni 2021 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 7. Oktober 2021 angenommen. Die mündliche Prüfung fand am 8. Oktober 2021 statt.

Abstract

In this thesis, we explore the efficiency of different direct discretization strategies to find numerical solutions to re-entry optimal control problems with minimum heating.

The first part of this thesis is dedicated to introducing the theoretical grounds of this work. We focus on SQP methods and the nonsmooth Newton method to solve nonlinear optimization problems numerically, and so-called *first discretize, then optimize* or direct discretization strategies for optimal control problems. The distinction between full and reduced discretization methods will be key to the development of different numerical strategies and the discussion on their efficiency in later chapters. The particularly sparse structure of the dynamics yielded by the application of the method of lines to PDE constrained optimal control problems is also illustrated here.

The second part of the thesis is dedicated to the models, definitions and approximations needed to compose atmospheric re-entry trajectory problems with thermodynamic constraints, in order to explore the application of our methodologies to different problems that may arise in this context.

In the third part of the thesis, the results of the application of a reduced discretization approach with the software OCPID-DAE1 to different re-entry optimal control problems are presented. The results show the robust applicability of this method to various problems featuring different scenarios, parameter optimization and coupled ODE-PDE problems.

In the last part of the thesis, our newly implemented strategy for the exploitaiton of the structure yielded by a full discretization approach with a nonsmooth Newton method and PDE discretization through the method of lines is presented. The efficiency of this strategy is demonstrated through its application to quadratic and nonlinear heat equation control problems. We consider as well the application of this method to a re-entry temperature control problem with a controllable active cooling system. The computational results show that while reduced discretization is a viable and robust option for smaller trajectory problems, a structure-exploiting method is necessary in order to tackle large PDE constrained optimal control problems.

Kurzzusammenfassung

In dieser Arbeit wird die Effizienz verschiedener direkter Diskretisierungsstrategien untersucht, um numerische Lösungen für die optimale Steuerung von Wiedereintrittsproblemen mit minimaler Erwärmung zu finden.

Im ersten Abschnitt dieser Arbeit werden die theoretischen Grundlagen für diese Arbeit vorgestellt. Wir interessieren uns für SQP-Methoden und nichtglatte Newton-Verfahren, sowie für so genannte *first discretize, then optimize* beziehungsweise direkte Diskretisierungsstrategien, um nichtlineare Optimierungsprobleme numerisch zu lösen. Die Differenzierung zwischen vollständiger und reduzierter Diskretisierung ist entscheidend für die Entwicklung verschiedener numerischer Ansätze und der anschließenden Betrachtung ihrer Effizienz in späteren Kapiteln. Die spezielle, dünn besetzte Struktur der Dynamik, die sich aus der Anwendung der Linienmethode auf Optimalsteuerungsprobleme mit PDE-Beschränkungen ergibt, wird hier ebenfalls dargestellt.

Der zweite Teil der Arbeit beschäftigt sich mit den Modellen, Definitionen und Approximationen, die für die Formulierung von atmosphärischen Wiedereintrittsproblemen mit thermodynamischen Beschränkungen erforderlich sind, um die Anwendung unserer Methoden auf verschiedene Probleme zu untersuchen, die in diesem Zusammenhang auftreten können.

Im dritten Abschnitt der Arbeit werden die Ergebnisse der Anwendung eines reduzierten Diskretisierungsansatzes mit Hilfe der Software OCPID-DAE1 auf verschiedene Wiedereintritts-Optimierungsprobleme präsentiert. Die Ergebnisse zeigen die robuste Anwendbarkeit dieser Methode auf verschiedene Probleme mit unterschiedlichen Szenarien, Parameteroptimierung und gekoppelten ODE-PDE-Problemen.

Schließlich wird im letzten Teil der Arbeit eine neue Strategie zur Ausnutzung der Struktur, die sich aus einer vollständigen Diskretisierung in Verbindung mit einer nicht-glatten Newton-Methode und einer PDE-Diskretisierung mittels der Linienmethode ergibt, vorgestellt. Die Effizienz dieses Ansatzes wird durch dessen Anwendung auf quadratische und nichtlineare Wärmeleitungsprobleme demonstriert. Außerdem analysieren wir die Anwendung dieser Methode auf ein Temperaturregelungsproblem beim Wiedereintritt mit einem steuerbaren aktiven Kühlsystem.

Die numerischen Ergebnisse zeigen, dass eine reduzierte Diskretisierung zwar eine praktikable und robuste Option für kleinere Bahnplanungsprobleme ist, andererseits aber eine Methode zur Ausnutzung der Struktur notwendig ist, um große, durch PDEs beschränkte Optimalsteuerungsprobleme zu bewältigen.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Professor Dr. Matthias Gerdts for giving me the opportunity to work on this topic, for his invaluable support and patience, and for always having his door open to answer my questions and offer new ideas and guidance.

I am grateful to Munich Aerospace for providing the financial support for this research, as well as numerous opportunities for exchange and collaboration. Being part of the Munich Aerospace community and the research group led by Professor Dr. Christian Mundt has been a great source of motivation and new ideas through these years, for which I am truly thankful.

I would also like to thank my colleagues from the Bundeswehr University who have accompanied me through this time. Special thanks go to Alberto De Marchi and Andreas Britzelmeier for their help and their suggestions to improve this work.

Finally, I would like to thank my family and friends for their support and encouragement throughout the last four years.

Contents

List of Figures

List of Tables

1 Introduction

Reducing the cost of access to space is one of the main current goals of the aerospace industry. The reduction of aerodynamic heating is vital for improving safety in flight and its derived costs, and most missions encounter their most extreme heating values during their re-entry phase. As a hypersonic vehicle descends into the atmosphere, it undergoes extreme convective heat rates ($\dot{q}_{conv}$) due to its high kinetic energy, and high efforts need to be made in order to prevent the internal structure of the vehicle from exceeding its limits. The Columbia disaster [96] dramatically demonstrated the fatal consequences of a failure to control extreme heating during re-entry. However, one does not need to consider extreme cases in order to see the benefit of minimizing heating in order to reduce the costs of aircraft design and other material resources. Therefore, finding appropriate strategies to reduce heat loads originated during re-entry can result in a great gain in safety and the possibility of a drastic reduction in costs.

This work is done in the framework of the interdisciplinary research project *Re-entry optimization to minimise heating or infrared signature* for Munich Aerospace e.V. - Bavarian Research Network. The main goal of the project was to model and optimize re-entry maneuvers for flight systems taking into account the heat flux, with a focus on safety aspects and efficiency. Some of the work produced in this project has been presented in [1, 2, 99, 100, 118, 119, 125, 126]. In particular, the contribution of this work is the development, analysis and comparison of efficient optimization methods for re-entry problems. Efficiency is always an important benefit to aim for when developing solvers for any kind of optimization problem, but it is also a key factor in the context of re-entry applications, where real-time trajectory computation and optimization are of high interest in order to produce systems adaptable to disturbances and configuration changes during flight. Furthermore, the high complexity, nonlinearity and dimension of aircraft trajectory and temperature evolution models can pose a big challenge to standard nonlinear optimizers, which makes the development of efficient methods able to deal with large, complex problems all the more necessary.

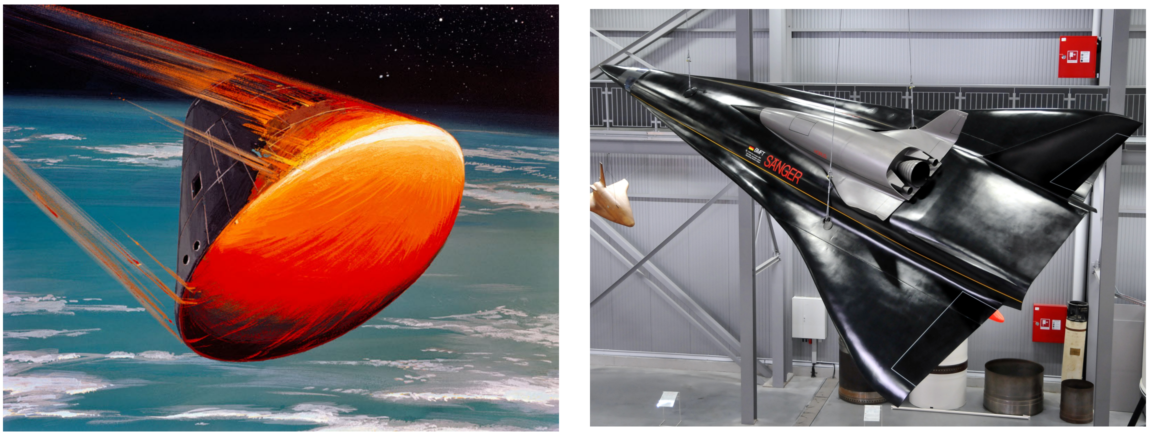

Figure 1.1: Apollo Command Module (left) and Sänger spacecraft (right). Image courtesy: NASA and Wikimedia Commons

There are four main variables that can contribute to reducing the heat load that an aircraft endures during re-entry: vehicle shape, Thermal Protection System (TPS) design, trajectory and active cooling systems. This work focuses mainly on the last two, i.e. finding optimal trajectories and cooling strategies to minimize heating, but TPS and shape design are also touched upon as additional proof of the versatility of our methodology to solve different problems related to re-entry. Furthermore, accurate modelling and optimization for TPS and vehicle shape require the application of computational fluid dynamics (CFD) techniques or other complicated calculations such as the heat transfer model by Fay and Riddell [33], which fall outside of this scope. Some in-depth studies on vehicle shape and TPS design optimization can be found in [27, 29, 70, 73, 82, 93, 136, 139]. There is extensive work in the topic re-entry trajectory and active cooling systems optimization with minimum heating, we offer a selection: [3, 4, 8, 21, 27, 28, 54, 71, 81, 88, 95, 104, 111, 140]. A good part of the literature focuses on the German Sänger concept [83] and the Apollo Command Module [60, 102], see Figure 1.1.

We base our approach to optimize re-entry problems in optimal control theory. Optimal control is the branch of applied mathematics that aims to derive a control for a given dynamical system in order to optimize a certain objective function. Optimal control problems are a generalization of variational problems where the control and state variables are separated and control constraints are admitted. This way, the dynamic behavior of the states can be described by dynamic equations, usually systems of ordinary and/or partial differential equations (ODEs and PDEs), that are influenced by the controls. There are applications of optimal control theory in many fields, including aerospace, robotics, bioengineering, economy, and other disciplines. One of the most important theoretical results of optimal control theory is Pontryagin's local minimum principle [108] that establishes necessary conditions for optimal control problems. A good overview of optimal control theory and techniques is gathered in [12, 41, 120].

In general, optimal control problems are not analytically solvable and require numerical methods in order to obtain an approximated solution. We follow a *first discretize,then optimize* approach in order to approximate the optimal control problem in question by a finite dimensional optimization problem with a suitable discretization scheme. In particular, a *direct discretization* approach aims to solve the finite optimization problem directly, as opposed to an *indirect discretizaton* that aims to solve the first order necessary optimality conditions with suitable techniques. The primary goal of this work is to test, analyze and compare in terms of efficiency different direct discretization methodologies to numerically solve re-entry optimal control problems with minimum heating. A reduced discretization method poses a small, dense optimization problem in terms of the discretized controls, while a full discretization method considers both the discretized states and controls as variables, yielding a large optimization problem with a sparse structure that needs to be exploited in order to produce solutions efficiently. This key difference poses the question of which method is more suitable and efficient for different types of optimal control problems, which is explored in this work in the context of re-entry.

A fully discretized optimal control problem can be solved using any suitable method for finite nonlinear optimization problems. The application of sequential quadratic programming (SQP) and interior point methods methods adapted to the sparse structure of fully discretized optimal control problems has been studied in [12, 13, 67, 137]. Several commercial and non-commercial software packages that use these methods to solve general sparse optimization problems are available, such as `SNOPT` [49, 50], `IPOPT` [135] and `WORHP` [18]. The use of nonsmooth Newton methods to solve the Karush-Kuhn-Tucker (KKT) necessary optimality conditions has been presented in [34, 72, 101, 123], and its application to discretized optimal control problems in [43, 44, 68, 75, 85]. Structure-exploiting strategies for nonsmooth Newton methods have been recently explored in [16, 25]. In order to take advantage of the structure generated by full discretization and the application of a certain nonlinear optimization method, it is necessary to use an optimizer tailored to sparse, large-scale problems, with an efficient LU decomposition method like MA48 from the HSL library [31] or a suitable subroutine from the LAPACK package [5] to solve the linear system in every iteration. A prior LU decomposition by blocks provides the opportunity to solve a series of smaller subsystems instead, which can also present an exploitable substructure. However, solvability of the linear system is only guaranteed under certain assumptions, which are recalled and explored in Chapter 2 for SQP and nonsmooth Newton methods, see [89] for details.

Optimal control problems involving PDEs are a challenging case of large-scale optimal control problems, especially when PDEs in several dimensions are present. They are treated theoretically in [63, 86, 127], and a good overview of numerical treatment of PDE optimal control problems can be found in [58, 64]. A simple but effective strategy to obtain numerical solutions is the method of lines; it consists in discretizing the PDE in space using finite differences [42, 47, 78, 107, 137], expressing it as a large ODE system. Like so, we can apply directly the methods used for standard ODE optimal control problems. Moreover, the doubly discretized problem in space and time presents a particularly sparse and banded structure, which makes a substructure exploitation particularly beneficial. Therefore, a full discretization approach with different structure exploitation strategies tailored to this type of problems is implemented and tested in this work for two-dimensional heat equations adapted from [57, 107], with an application to active cooling control during re-entry. Similar work on structure exploitation for discretized PDE constrained problems can be found in [47, 59, 103, 137].

Furthermore, the combination of aerodynamic trajectory dynamics with a temperature evolution model generates a complex case of a coupled ODE-PDE optimal control problem, where the goal is to minimize the temperature increase by controlling the trajectory. Coupled problems are a relatively recent object of study, and general theoretical results are not available; however, there is a great deal of literature focused on the analysis and derivation of first order necessary optimality conditions for specific problems, see [20, 32, 65, 76, 105, 138] for some examples, and numerical solutions for different applications are calculated in [21, 66, 78, 79, 140]. While a full discretization approach seems to be the most efficient choice in the case of PDE constrained problems, a reduced discretization approach can also produce numerical solutions robustly and efficiently for PDEs of smaller dimensions and in the absence of a large number of controls. The described coupled ODE-PDE trajectory problem falls under this category, which makes the application of this method to re-entry trajectory problems also worth exploring. This is done in this work using OCPID-DAE1 [40], an optimizer of the Institute for Applied Mathematics and Scientific Computing at the Universität der Bundeswehr München by Prof. Matthias Gerdts. OCPID-DAE1 solves general optimal control problems featuring differential algebraic equations (DAEs) using multiple shooting for integration with an SQP method for general constrained nonlinear problems, and it has been successfully used to solve a variety of optimal control problems and coupled ODE-PDE problems, see some examples in [17, 45, 77, 78, 87, 92, 99, 112].

The main contribution of this thesis, as presented in Chapter 5 and in the article [100], is the implementation of a structure and substructure exploitation strategy for fully discretized optimal control problems solved with the nonsmooth Newton method. We provide an analysis and comparison of the application of this methodology and a reduced discretization approach to different optimal control problems in the context of re-entry with minimum heating, focusing on efficiency and the limitations of their applicability. Our method proves to be an efficient alternative to the reduced discretization approach for PDE constrained optimal control problems, and it could be used for general large-scale optimal control problems. We present as well several alternatives for structure and substructure exploitation with different linear solvers, and compare their efficiency on several large-scale problems involving discretized heat equations through the method of lines. A discussion on the numerical complications that may arise related to the solvability of the linear systems that need to be solved in the nonsmooth Newton method is also included.

This thesis is structured as follows. In Chapter 2, we present some general results and definitions from nonlinear optimization and optimal control theory. We first recall the basic concepts and the KKT conditions, and then we go into detail about SQP methods and nonsmooth Newton methods, providing some convergence and solvability results for the linear systems that arise from their application. In the second part of the chapter, we focus on general optimal control problems, the necessary optimality conditions established by the local minimum principle, and the direct discretization approach to find numerical solutions. We describe in detail the reduced and full discretization approaches, presenting an adaptation of the discrete local minimum principle for general one-step methods. Finally, the method of lines to discretize PDEs and transform them into a sparse ODE system is presented.

Chapter 3 gives an outline of the various models used to define the different re-entry problems that are solved numerically in this thesis. There are several aspects to consider that are presented in detail: an atmospheric model to obtain air temperature, density and other quantities at different altitudes; a dynamic model for the position and direction of the aircraft, taking into account the aerodynamic and gravitational forces to define a system of equations of motion; a thermodynamic model involving the heat equation to calculate temperature evolution, with models for external and internal heating and a model for the interaction between the TPS and the active cooling system; and finally, a parametric shape model for an Apollo-type capsule.

Chapter 4 is dedicated to finding optimal re-entry trajectories with minimum heating using a reduced discretization approach with the software package OCPID-DAE1, expanding from the results presented in [99]. We formulate re-entry trajectory problems in different scenarios and with additional challenges, such as the aforementioned parametric capsule shape optimization and a coupled ODE-PDE system to calculate the temperature in a one-dimensional section through the TPS. For the last one, we consider as well different objective functions based on the temperature and the heat flux. Numerical solutions are depicted for all of the defined problems.

In Chapter 5, our structure and substructure exploitation strategies for fully discretized optimal control problems solved with the nonsmooth Newton method are presented. Different approaches to solving the obtained linear system are considered and compared computationally on a benchmark quadratic PDE problem involving a two-dimensional heat equation. The approaches that yield the better results in terms of computational time are tested further on a nonlinear PDE benchmark problem, and numerical solutions and particularities of each approach are discussed. These results are taken and expanded upon from the publication [100]. To conclude the numerical results, a re-entry temperature control problem with a controllable active cooling system is considered and solved numerically with this methodology. Finally, the suitability and limitations of the reduced and full discretization approaches based on the obtained computational results are discussed.

2 Optimal control and optimization theory

In this chapter, we remind the reader of a few basic nonlinear optimization concepts and methods used for finding numerical solutions, we define the optimal control framework, and we recall some of the most important concepts and results of optimal control theory. We recall as well a brief classification of the methods typically used to approach optimal control methods, both analytically and numerically, and then focus on a *first discretize, then optimize* or direct discretization approach. Finally, a discretization strategy for the heat equation with the method of lines, which is used in the following chapters to transform PDE constrained optimal control problems into regular ODE constrained ones, is described.

Although both nonlinear optimization and optimal control theory are vast fields, we focus on the optimization methods and techniques to solve optimal control problems numerically that will be used in this scope. We provide as well the theoretical results that can be relevant to better understand the strategies and numerical results presented in the following chapters; for example, the discretized minimum principle provides a useful relation between the theory applied to the discretized problems and their continuous versions. An issue that comes up when optimizers fail is the solvability and conditioning of the linear systems that arise and therefore, a short outline of some sufficient conditions is given for the presented optimization methods that will be useful for the discussion of the results in Chapter 5.

2.1 Nonlinear optimization

We consider nonlinear optimization problems of the general form:

Problem 2.1 (NLP)**.** *Minimize $f(z)$ subject to the constraints*

$$g(z) \leq 0, \tag{2.1}$$

$$h(z) = 0, \tag{2.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be twice continuously differentiable functions. Function $f(z)$ is referred to as the *objective function*, (2.1) are referred to as the *inequality constraints* and (2.2) are referred to as the *equality constraints*. It will be useful for the following development to define as well the *feasible set*:

$$\Omega := \{z \in \mathbb{R}^n \mid g_i(z) \leq 0,\ i = 1, ..., m,\ h_j(z) = 0,\ j = 1, ..., p\},$$

as well as the active set of the inequality constraints for a certain $z \in \mathbb{R}^n$:

$$\mathcal{A}_g(z) := \{i \in 1, ..., m \mid g_i(z) = 0\}.$$

With the goal of trying to define solutions to (NLP), we can define a *global minimum* as a $z^* \in \Omega$ such that

$$f(z^*) \leq f(z), \qquad \forall z \in \Omega,$$

and $z^* \in \mathbb{R}^n$ is a *local minimum* if there exists an $\epsilon > 0$ such that

$$f(z^*) \leq f(z), \qquad \forall z \in \Omega \cap \mathcal{U}_\epsilon(z^*).$$

Before establishing the main result that we want to reproduce here, the well-known *Karush-Kuhn-Tucker conditions* (KKT conditions), we mention some of the constraint qualifications that are required to guarantee that the conditions hold. The most common ones are *Linear Independence Constraint Qualification (LICQ)*, which holds for a given feasible $z \in \Omega$ if the gradients

$$\{\nabla g_i(z),\ i \in \mathcal{A}_g(z)\} \cup \{\nabla h_j(z),\ j = 1, ..., p\} \tag{2.3}$$

are linearly independent, or the *Mangasarian-Fromowitz Constraint Qualification (MFCQ)*, which holds if the gradients

$$\{\nabla h_j(z),\ j = 1, ..., p\}$$

are linearly independent and there exists a vector $d \in \mathbb{R}^n$ such that

$$\nabla g_i(z)^\top d < 0,\ i \in \mathcal{A}_g(z)$$
$$\nabla h_j(z)^\top d = 0,\ j = 1, ..., p$$

We define as well the *Lagrangian* or *Lagrange function* for (NLP), given vectors $\lambda \in \mathbb{R}^p$, $\mu \in \mathbb{R}^m$, as:

$$L(z, \lambda, \mu) = f(z) + \lambda^\top h(z) + \mu^\top g(z). \tag{2.4}$$

The vectors λ and μ are often referred to as *Lagrange multipliers*. One of the most fundamental results in nonlinear programming is the KKT conditions, established by [74, 84], since they provide first order necessary optimality conditions for (NLP):

Theorem 2.1 (KKT conditions). *Let z^* be a local minimum of Problem 2.1 satisfying a constraint qualification. Then there exist Lagrange multipliers $\lambda^* \in \mathbb{R}^p$ and $\mu^* \in \mathbb{R}^m$ such that*

$$\nabla_z L(z^*, \lambda^*, \mu^*) = 0, \tag{2.5}$$
$$h(z^*) = 0 \tag{2.6}$$
$$\mu^* \geq 0,\ g(z^*) \leq 0,\ \mu_i^* g_i(z^*) = 0, \quad i = 1, .., m. \tag{2.7}$$

The most commonly used methods to solve constrained nonlinear optimization problems are based on constructing sequences $\{(z^k, \lambda^k, \mu^k)\}$ iteratively that eventually converge to a point that satisfies the necessary KKT conditions. In the case in which inequality constraints are not present, it suffices to solve the system of nonlinear equations (2.5) - (2.6) with the Newton method, which is referred to as the Lagrange-Newton Method. Sequential Quadratic Programming (SQP) methods can be considered an extension of of the Lagrange-Newton Method based on finding solutions to a sequence of Quadratic Programming (QP) subproblems. Interior Point methods add slack variables to the inequality constraints and attempt to solve a series of barrier problems that converges to a feasible solution of the problem. The nonsmooth Newton method introduces a complementarity function that becomes 0 if and only if condition (2.7) is satisfied.

We focus on SQP methods and the nonsmooth Newton method in the following sections, and we introduce some of the theoretical results that guarantee their convergence. It is important to acknowledge that none of these approaches can be considered superior to the others from a purely mathematical point of view. However, their practical performance depends largely on the implementation of the method and its strategies to deal with the numerical challenges that may arise. Moreover, some implementations might be better for some problems than others in terms of convergence, efficiency and robustness.

For more in-depth theory on nonlinear programming, the reader can consult the classical textbooks [10, 98].

2.1.1 SQP method

SQP methods were first developed in [52, 56, 109, 117] and since then, they have become one of the most powerful iterative algorithms to solve nonlinear problems. We define priorly the Hessian of the Lagrange function at a point (z^k, λ^k, μ^k) as

$$H_k := L''_{zz}(z^k, \lambda^k, \mu^k)$$

In order to find a solution to (NLP) iteratively, the following quadratic optimization problem is considered:

Problem 2.2 (QP)**.** *Minimize*

$$\frac{1}{2} d^\top H_k d + \nabla f(z^k)^\top d$$

with respect to $d \in \mathbb{R}^n$ *subject to the constraints*

$$\begin{aligned} g(z^k) + g'(z^k)d &\leq 0, \\ h(z^k) + h'(z^k)d &= 0. \end{aligned}$$

This problem arises as a linearization of the KKT conditions: a search direction d is computed in order to minimize a quadratic approximation of the Lagrangian subject to a linear approximation of the constraints. The KKT conditions for (QP) would be

$$\begin{aligned} H_k d + \nabla f(z^k) + g'(z^k)^\top \mu + h'(z^k)^\top \lambda &= 0, \\ h(z^k) + h'(z^k)d &= 0, \\ g(z^k) + g'(z^k)d \leq 0, \quad \mu \geq 0, \quad \mu^\top (g(z^k) + g'(z^k)d) &= 0, \end{aligned}$$

being $\mu \in \mathbb{R}^m, \lambda \in \mathbb{R}^p$ the Lagrange multipliers for the inequality and equality constraints of (QP), respectively. A local SQP method would read as:

Algorithm 2.1. (Local SQP Method)

(0) Choose (z^0, λ^0, μ^0) and set $k = 0$.

(1) If (z^k, λ^k, μ^k) is a KKT point of (NLP), STOP.

(2) Compute a solution $(d^k, \lambda^{k+1}, \mu^{k+1})$ of (QP).

(3) Set $z^{k+1} = z^k + d^k$, $k = k + 1$ and go to (1).

The following result from [7, Theorem 7.5.4] establishes the convergence of Algorithm 2.1 to a local minimum of (NLP) under certain conditions:

Theorem 2.2 (Local convergence of the SQP method)**.** *Let the following assumptions hold:*

(a) Let z^* be a local minimum of (NLP).

(b) Let f, g and h be twice continuously differentiable with Lipschitz continuous second derivatives.

(c) Let (LICQ) (2.3) hold at z^*.

(d) Let

$$v^\top L''_{zz}(z^*, \lambda^*, \mu^*)v > 0$$

for all $v \in \mathbb{R}^n$, $v \neq 0$ with

$$g'_i(z^*)v = 0, \ i \in \mathcal{A}_g(z^*), \quad h'_j(z^*)v = 0, \ j = 1, ..., p.$$

Then there exist neighborhoods U of (z^*, λ^*, μ^*) and V of $(0, \lambda^*, \mu^*)$ such that all quadratic optimization problems (QP) have a unique local solution d^k with unique multipliers μ^{k+1} and λ^{k+1} in V for every (z^0, λ^0, μ^0) in U. Moreover, the sequence (z^k, λ^k, μ^k) converges locally quadratically to (z^*, λ^*, μ^*).

Therefore, finding a local minimum of (NLP) is reduced to solving a series of quadratic problems for which there are powerful algorithms available using primal or dual active-set methods, interior point methods or semismooth Newton methods [24, 46, 48, 51, 55, 130]. We selected the primal active-set methods given in [7] for this work, so we expand briefly on them. Given a solution d^* to (QP) with active-set $\mathcal{A}_G(d^*) \subseteq \mathcal{I} := \{1, ..., m\}$, where $G(d) := g(z^k) + g'(z^k)d$, and $\mathcal{J} := \{1, ..., p\}$, we define the quadratic problem for iteration k with the general form:

Problem 2.3 (Active-set QP)**.** *Minimize*

$$\frac{1}{2} d^\top Q d + c^\top d$$

with respect to $d \in \mathbb{R}^n$ *subject to the constraints*

$$\begin{aligned} a_i^\top d &= v_i, \quad i \in \mathcal{A}_G(d^*), \\ b_j^\top d &= w_j, \quad j \in \mathcal{J}, \end{aligned}$$

where $Q \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^n$, $a, v \in \mathbb{R}^m$, $b, w \in \mathbb{R}^p$. Since the active set at the solution d^* is not known a-priori, the idea of active-set methods is to estimate $\mathcal{A}_G(d^*)$ iteratively through a series of sets $\mathcal{A}_k \subseteq \mathcal{I}$ by solving the auxiliary problem for a given feasible d:

Problem 2.4 (Auxiliary QP)**.** *Minimize*

$$\frac{1}{2}(d + \Delta d)^\top Q (d + \Delta d) + c^\top (d + \Delta d)$$

with respect to $\Delta d \in \mathbb{R}^n$ *subject to the constraints*

$$\begin{aligned} a_i^\top \Delta d &= 0, \quad i \in \mathcal{A}_k, \\ b_j^\top \Delta d &= 0, \quad j \in \mathcal{J}. \end{aligned}$$

Given the absence of inequality constraints, this problem can be solved by solving the linear system given by the KKT conditions:

$$\begin{pmatrix} Q & A_{\mathcal{A}_k}^\top & B^\top \\ A_{\mathcal{A}_k} & 0 & 0 \\ B & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta d \\ \mu_{\mathcal{A}_k} \\ \lambda \end{pmatrix} = \begin{pmatrix} -(Qd + c) \\ 0 \\ 0 \end{pmatrix}, \tag{2.8}$$

where $A_{\mathcal{A}_k} := (a_i^\top)_{i \in \mathcal{A}_k}$, $B := (b_j^\top)_{j \in \mathcal{J}}$, and μ, λ are the multipliers associated to the inequality and equality constraints in (QP), respectively. Hence, the algorithm to solve (QP) with the active-set method would read as:

Algorithm 2.2. (Active-set method, [7])

(0) Choose a feasible initial guess d^0 and set $k = 0$, $\mathcal{A}_0 = \mathcal{A}(d^0)$.

(1) Compute a solution $(\Delta d, \mu_{\mathcal{A}_k}, \lambda)$ of (2.8).

(2) If $\Delta d = 0$ and $\mu_{\mathcal{A}_k} \geq 0$, set $\mu_i = 0$ for all $i \in \mathcal{I} \setminus \mathcal{A}_k$, STOP.

(3) If $\Delta d = 0$ and $\mu_q := \min\{\mu_i \mid i \in \mathcal{A}_k\} \leq 0$, set $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{q\}$, $k := k+1$ and go to (2).

(4) If $\Delta d \neq 0$ and $a_i^\top (d^k + \Delta d) \leq u_i$ for all $i \in \mathcal{I} \setminus \mathcal{A}_k$, set $d^{k+1} := d^k + \Delta d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$, $k := k+1$ and go to (2).

(5) Find $r \in \mathcal{I} \setminus \mathcal{A}_k$ with $a_r^\top \Delta d > 0$ and

$$t_k := \frac{v_r - a_r^\top d^k}{a_r^\top \Delta d} = \min\left\{ \frac{v_i - a_i^\top d^k}{a_i^\top \Delta d} \mid i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^\top \Delta d > 0 \right\}.$$

Set $d^{k+1} := d^k + t_k \Delta d$, $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{r\}$, $k := k+1$ and go to (2).

Note that finding a feasible initial guess is not always trivial. A possible way to do so is to solve a relaxed version of (QP) by introducing a slack variable to relax the constraints whose value is introduced with a penalty parameter in the objective function [109].

Since Theorem 2.2 only guarantees convergence for starting points in some neighborhood of a local minimum of (NLP), we need to use so-called globalization strategies so that the method converges for arbitrary starting values. This is achieved by introducing a step $\alpha_k \in \mathbb{R}$ and choosing

$$z^{k+1} = z^k + \alpha_k d^k \tag{2.9}$$

as new iterate. The step size is computed by minimizing a certain merit function that measures the improvement of the iterates along the direction d^k depending on a penalty parameter $\nu > 0$ (or several). Typically used merit functions are:

- The L_1 penalty function (non-differentiable, [109]):

$$L_1(z; \nu) = f(z) + \nu \sum_{i=1}^{m} \max\{0, g_i(z)\} + \nu \sum_{i=1}^{p} |h_i(x)|.$$

- The augmented Lagrangian function ([117]):

$$L_a(z, \lambda, \mu; \nu) = f(z) + \lambda^\top h(z) + \frac{\nu}{2} \sum_{i=1}^{p} h_i(z)^2$$
$$+ \frac{1}{2\nu} \sum_{i=1}^{m} ((\max\{0, \mu_i + \nu g_i(z)\})^2 - \mu_i^2)$$

Under certain regularity conditions and an adequate choice of $\nu > 0$, it holds that every local minimum z^* of (NLP) is also a local minimum of these functions and therefore, for any direction of descent d for (NLP) it holds i.e. for the directional derivative of L_1

$$L_1'(z; d; \nu) < 0.$$

In order to determine the step α_k that minimizes the merit function for the search direction d^k in each step to compute the next iterate (2.9), we use the so-called Armijo line-search. Using L_1 as merit function, we define the function

$$\varphi(\alpha) := L_1(x^k + \alpha d^k; \nu).$$

Given that

$$\varphi'(0) = L_1'(x^k; d^k; \nu) < 0,$$

we try to find the largest α for which the *Armijo conditon* holds:

$$\varphi(\alpha) \leq \varphi(0) + \sigma \ \alpha \ \varphi'(0)$$

for a certain $\sigma \in (0, 1)$. To this end, we select $\beta \in (0, 1)$ and perform a line-search by reducing α iteratively with

$$\alpha = \beta^j$$

for $j \in \mathbb{N}_0$ until the Armijo condition holds for α. Therefore, a global version of Algorithm 2.1 with an Armijo line-search and L_1 as merit function reads as:

Algorithm 2.3. (Global SQP Method, [7])

(0) Choose (z^0, λ^0, μ^0), $\beta \in (0, 1)$, $\sigma \in (0, 1)$ and set $k = 0$.

(1) If (z^k, λ^k, μ^k) is a KKT point of (NLP), STOP.

(2) Compute a solution $(d^k, \lambda^{k+1}, \mu^{k+1})$ of (QP).

(3) Adapt the penalty parameter ν.

(4) Armijo line-search: determine a step size $\alpha_k = \max_{j \in \mathbb{N}_0} \beta^j$ such that

$$L_1(z^k + \beta^j d^k; \nu) \leq L_1(z^k; \nu) + \sigma \alpha_k L_1'(z^k; d^k; \nu)$$

(5) Set $z^{k+1} = z^k + \alpha_k d^k$, $k = k + 1$ and go to (1).

Strategies for adapting penalty parameters, calculating the derivatives of L_1 and L_a and global convergence results for Algorithm 2.3 using these merit functions can be found in [56, 109, 117]. There are other alternatives for globalization: filter methods that evaluate search directions on their improvement for the objective function and the constraint violation separately, see [38, 39, 129], and trust-region methods that limit the search direction to a certain trust-region radius, adjusted in each iteration according to the reduction in a merit function and a model function, see [23, 37].

The bulk of the numerical calculations and computational effort in Algorithm 2.3 lie in solving linear system (2.8) iteratively in order to find a solution of (QP) in step (2) for every iteration. The solvability of this system is not always guaranteed, specially when the starting point is chosen poorly, but there are certain techniques to deal with potentially singular matrices or inconsistent linear systems, see [53]. We examine nonetheless the conditions established by Theorem 2.2 for the following matrix:

$$K := \begin{pmatrix} H & E^\top & F^\top \\ E & 0 & 0 \\ F & 0 & 0 \end{pmatrix}, \quad H := L''_{zz}(z, \lambda, \mu),\ E := \{\nabla g_i(z)^\top\}_{i \in \mathcal{A}_g(z)},\ F := \nabla h(z)^\top \tag{2.10}$$

for a $(z, \mu, \lambda) \in \mathbb{R}^{n+m+p}$, often referred to as the *KKT matrix*. The matrix in (2.8) differs from K only if the set of active inequality constraints for (NLP) and (QP) differ; however, in the range of convergence the active set should not change, which means that locally around the solution the two matrices should be the same. Full rank of K at the solution is required by the conditions of Theorem 2.2 as established by the following Lemma from ([89]):

Lemma 2.1 ([89], Lemma A.3)**.** *For $n, m \in \mathbb{N}$, $n \geq m$, suppose $C \in \mathbb{R}^{n \times n}$ is symmetric and $D \in \mathbb{R}^{m \times n}$ has full rank m. Moreover, it holds*

$$d^\top C d > 0 \quad \text{for all } d \in \ker(D) \setminus \{0\}.$$

Then, the matrix

$$\begin{pmatrix} C & D^\top \\ D & 0 \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$$

is non-singular.

Corollary 2.1. *If conditions (c) and (d) from Theorem 2.2 hold, the KKT matrix K is non-singular.*

Proof. $D := \begin{pmatrix} E \\ F \end{pmatrix}$ has full rank due to the (LICQ) condition established by (c), and $C := H$ verifies $d^\top H d > 0$ for all $d \in \ker(D) \setminus \{0\}$ as established by (d), so the conditions of Lemma 2.1 apply. □

Note that (LICQ) only requires for the gradients of the active inequality constraints and the equality constraints to be linearly independent. This allows, for example, for box constraints of the type

$$a \leq g(z) \leq b$$

with $a, b \in \mathbb{R}^m, a < b$ to be included in (NLP). We consider the following problem:

Problem 2.5 (NLP with box constraints)**.** *Minimize $f(z)$ subject to the constraints*

$$\tilde{g}(z) \leq 0, \tag{2.11}$$

$$h(z) = 0, \tag{2.12}$$

where

$$\tilde{g}(z) = \begin{pmatrix} a - g(z) \\ g(z) - b \end{pmatrix}$$

with f, g, h as defined for (NLP) and $\tilde{g} : \mathbb{R}^n \to \mathbb{R}^{2m}$ continuously differentiable by extension. We have that

$$\begin{aligned} \tilde{g}_i(z) &= a_i - g_i(z), \\ \tilde{g}_{i+m}(z) &= g_i(z) - b_i. \end{aligned}$$

The set of all the gradients of the constraints for Problem 2.5 is:

$$\begin{aligned} &\{\nabla \tilde{g}_i(z),\ i = 1, ..., 2m\} \cup \{\nabla h_j(z),\ j = 1, ..., p\} = \\ &\{-\nabla g_i(z),\ i = 1, .., m\} \cup \{\nabla g_i(z),\ i = 1, ..., m\} \cup \{\nabla h_j(z),\ j = 1, ..., p\}, \end{aligned} \tag{2.13}$$

which are linearly dependent for all $z \in \mathbb{R}^n$ if $m \geq 1$, regardless of g and h. However, defining the partition of $\mathcal{A}_{\tilde{g}}(z)$:

$$I_1 = \{i \in \mathcal{A}_{\tilde{g}}(z) : g_i(z) = a_i\}, \quad I_2 = \{i \in \mathcal{A}_{\tilde{g}}(z) : g_i(z) = b_i\},$$

we have that $I_1 \cap I_2 = \emptyset$ since $a_i < b_i$ and if $i \in \mathcal{A}_{\tilde{g}}(z)$, then either $g_i(z) = a_i$ or $g_i(z) = b_i$, so $\mathcal{A}_{\tilde{g}}(z) = I_1 \uplus I_2$. Therefore,

$$\begin{aligned} &\{\nabla \tilde{g}_i(z), i \in \mathcal{A}_{\tilde{g}}(z)\} \cup \{\nabla h_j(z),\ j = 1, ..., p\} \\ &= \{-\nabla g_i(z), i \in I_1\} \cup \{\nabla g_i(z), i \in I_2\} \cup \{\nabla h_j(z),\ j = 1, ..., p\}. \end{aligned}$$

which can be linearly independent at the solution, as opposed to (2.13).

We finish this section with a note on how to calculate or estimate numerically the Hessians of the Lagrange function H_k for every iteration. Using the exact Hessian matrix, when explicitly known, comes with the risk of H_k being indefinite, which poses a problem to find a solution to (QP) since it is no longer convex. However, it also comes with the advantage of being able to exploit the structure of the matrix, which is useful and even necessary with large and sparse problems. In the case of dense problems, it is preferable to use BFGS updates [109] to produce positive definie approximations to H_k. Starting with a symmetrical, positive-definite H_0 (i.e. the identity matrix), the following Hessians $H_k,\ k = 1, 2, ...$ are calculated with the formula

$$H_{k+1} = H_k + \frac{q^k (q^k)^\top}{(q^k)^\top d^k} - \frac{H_k d^k (H_k d^k)^\top}{(d^k)^\top H_k d^k},$$

where

$$
\begin{aligned}
q^k &= \theta_k y^k + (1-\theta_k) H_k d^k, \\
y^k &= \nabla_x L(x^{k+1}, \lambda^k, \mu^k) - \nabla_x L(x^k, \lambda^k, \mu^k), \\
\theta_k &= \begin{cases} 1, & \text{if } (d^k)^\top y^k \geq 0.2 (d^k)^\top H_k d^k, \\ \frac{0.8(d^k)^\top H_k d^k}{(d^k)^\top H_k d^k - (d^k)^\top y^k}. & \text{otherwise} \end{cases}
\end{aligned}
$$

2.1.2 Nonsmooth Newton method

A nonsmooth Newton method [110] is an extension of Newton's method for solving a nonlinear equation of several variables to a nonsmooth case by using the generalized Jacobian instead of the derivative. Following the approach from [34], it can be used for solving nonlinear programs by finding a solution to the KKT conditions (2.5) - (2.7) using a complementarity function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ that satisfies:

$$
a \geq 0, \; b \geq 0, \; ab = 0 \quad \Longleftrightarrow \quad \varphi(a,b) = 0,
$$

for all $a, b \in \mathbb{R}$. Using a function of these characteristics, the conditions (2.5)-(2.7) are converted to the nonlinear equation system

$$
F(Z^*) = \begin{pmatrix} \nabla_z L(z^*, \lambda^*, \mu^*) \\ h(z^*) \\ \varphi(-g_1(z^*), \mu_1^*) \\ \vdots \\ \varphi(-g_m(z^*), \mu_m^*) \end{pmatrix} = 0, \qquad Z^* = (z^*, \lambda^*, \mu^*),
$$

where continuous differentiability is not necessarily assumed from F. A possible choice for φ that offers several advantages such as local Lipschitz continuity is the Fischer-Burmeister function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$
\varphi(a,b) := \sqrt{a^2 + b^2} - a - b. \tag{2.14}
$$

Because the Fischer-Burmeister function φ is not differentiable at $(a,b) = (0,0)$ (but continuously differentiable everywhere else), F is only differentiable in

$$
D_F := \{ Z = (z, \lambda, \mu)^\top \in \mathbb{R}^{n+m+p} \mid |g_i(z)| + |\mu_i| > 0, \; i = 1, ..., m \}.
$$

Since φ is locally Lipschitz continuous, F is locally Lipschitz continuous as well and therefore, F is differentiable almost everywhere by Rademacher's theorem and the B(ouligand)-Differential

$$\partial_B F(Z) := \left\{ V \;\middle|\; V = \lim_{\substack{Z_i \in D_F \\ Z_i \to Z}} F'(Z_i) \right\}$$

is well defined, and its convex hull

$$\partial F(Z) := conv(\partial_B F(Z))$$

is Clarke's generalized Jacobian, which is non-empty, convex and compact [22]. In the case of the Fischer-Burmeister function, the generalized Jacobian of φ is defined by

$$\partial\varphi(a,b) = \begin{cases} \left\{ \left(\frac{a}{\sqrt{a^2+b^2}} - 1, \frac{b}{\sqrt{a^2+b^2}} - 1 \right) \right\}, & if\ (a,b) \neq (0,0), \\ \{(s,t) \mid (s+1)^2 + (t+1)^2 \leq 1\}, & if (a,b) = (0,0). \end{cases}$$

Given the existence of the generalized Jacobian, the nonsmooth Newton method can be applied to solve the equation $F(Z) = 0$. The procedure is identical to the classical Newton method, with the exception that some element of the B-differential replaces any non-existing Jacobian that may arise. The algorithm to solve (NLP) would read as:

Algorithm 2.4. (Local Nonsmooth Newton Method)

(0) Choose Z^0 and set $k = 0$.

(1) If some stopping criterion is satisfied, STOP.

(2) Choose $V_k \in \partial F(Z^k)$ and compute the search direction d^k as the solution of the linear equation

$$V_k d^k = -F(Z^k).$$

(3) Set $Z^{k+1} = Z^k + d^k$, $k = k + 1$ and go to (1).

We reproduce here a result from [110] on the local convergence of this algorithm. It introduces the new notion of semismoothness, but it suffices to know, for our purpose, that the Fischer-Burmeister function is semismooth [35].

Theorem 2.3 (Local convergence of the nonsmooth Newton method)**.** *Suppose that Z^* is a solution of $F(Z) = 0$, F is locally Lipschitz continuous and semismooth at Z^*, and all $V \in \partial F(Z^*)$ are non-singular. Then Algorithm 2.4 is well-defined and convergent to Z^* in a neighborhood of Z^*.*

The local nonsmooth Newton method needs to be globalized in order to obtain global convergence for arbitrary starting points Z^0. Several globalizations for semismooth Newton methods have been presented and explored in the literature, such as trust-region or primal-dual active-set strategies [68, 75, 128]. We focus on a line search globalization [43, 72] with the merit function

$$\Theta(Z) := \frac{1}{2}\|F(Z)\|^2. \tag{2.15}$$

Given that the squared Fischer-Burmeister function φ^2 is continuously differentiable, $\Theta(Z)$ is continuously differentiable as well, and it holds

$$\nabla\Theta(Z) = V^\top F(Z),$$

where $V \in \partial F(Z)$. Thus, for any search direction d obtained by solving $Vd = -F(Z)$ it holds

$$\nabla\Theta(Z)^\top d = -F(Z)^\top F(Z) = -\|F(Z)\|^2 = -2\Theta(Z).$$

Therefore, d is a direction of descent if $F(Z) \neq 0$ and Armijo's line search is well-defined. We can define a global version of Algorithm 2.4 as:

Algorithm 2.5. (Globalized Nonsmooth Newton Method)

(0) Choose Z^0, $\beta \in (0,1)$, $\sigma \in (0, 1/2)$ and set $k = 0$.

(1) If some stopping criterion is satisfied, STOP.

(2) Choose $V_k \in \partial F(Z^k)$ and compute the search direction d^k as the solution of the linear equation

$$V_k d^k = -F(Z^k). \tag{2.16}$$

(3) Armijo line-search: determine a step size $\alpha_k = \max_{j\in\mathbb{N}} \beta^j$ such that

$$\Theta(Z^k + \alpha_k d^k) \leq \Theta(Z^k) + \sigma\alpha_k \nabla\Theta(Z^k)^\top d^k.$$

(4) Set $Z^{k+1} = Z^k + \alpha_k d^k$, $k = k+1$ and go to (1).

Superlinear convergence of Algorithm 2.5 is guaranteed by this result from [72] under certain solvability assumptions:

Theorem 2.4 (Global convergence of the nonsmooth Newton method)**.** *Suppose that equation (2.16) is solvable for each k and Z^* is an accumulation point of $\{Z^k\}$ generated by Algorithm 2.5. Then Z^* is a solution of $F(Z) = 0$ and $\{Z^k\}$ converges to Z^* superlinearly if $V^* \in \partial F(Z^*)$ is non-singular.*

As we did for the KKT matrix K (2.10) in the previous section, we address now the solvability of system (2.16) i.e. the non-singularity of matrix

$$V(Z) := \begin{pmatrix} H & E^\top & F^\top \\ -SE & T & 0 \\ F & 0 & 0 \end{pmatrix} \in \partial F(Z), \tag{2.17}$$

$$H = L''_{zz}(z,\lambda,\mu),\ E = \nabla g(z)^\top,\ F = \nabla h(z)^\top, \tag{2.18}$$

$$S = diag((s_1, ..., s_m)),\ T = diag((t_1, ..., t_m)),\ (s_i, t_i) \in \partial\varphi(-g_i(z), \mu_i) \tag{2.19}$$

for an arbitrary $Z = (z, \lambda, \mu) \in \mathbb{R}^{n+m+p}$, as required by Theorem 2.4 for the iterates Z^k. We define the following partition of $I := \{1, ..., m\}$:

$$J := \{i \in I : g_i(z) = 0\}, \quad I \setminus J = \{i \in I : g_i(z) \neq 0\}$$

We cite as well some properties from the Fischer-Burmeister function that can be found in [34] or derived directly from its definition:

(F1) $(s_i, t_i) \neq (0,0)$ for all $i \in I$.

(F2) $t_i = 0$ if and only if $i \in J$.

(F3) $(s_i, t_i) \in [-2,0] \times [-2,0]$, which means $s_i \leq 0, t_i \leq 0$ for all $i \in I$.

We define the matrices:

$$E_J := (\nabla g_i(z)^\top)_{i \in J}, \quad E_{I\setminus J} := (\nabla g_i(z)^\top)_{i \in I\setminus J}, \tag{2.20}$$

$$\bar{H} := H + \sum_{i \in I\setminus J} \frac{s_i}{t_i} \nabla g_i(z) \nabla g_i(z)^\top = H + diag\left(\left(\frac{s_i}{t_i}\right)_{i \in I\setminus J}\right) E_{I\setminus J}^\top E_{I\setminus J}, \tag{2.21}$$

$$\bar{K} := \begin{pmatrix} \bar{H} & E_J^\top & F^\top \\ E_J & 0 & 0 \\ F & 0 & 0 \end{pmatrix}. \tag{2.22}$$

The following theorem adapted from [34], where it is proved in absence of equality constraints, links the non-singularity of $V(Z)$ to the non-singularity of the matrix $\bar{K}$, similar to the KKT matrix K:

Theorem 2.5 ([34], Theorem 4.1)**.** *Let $Z = (z, \lambda, \mu) \in \mathbb{R}^{n+m+p}$. The matrix $V(Z)$ is non-singular if and only if $\bar{K}$ is non-singular.*

Proof. Considering the homogenous system

$$V(Z) \begin{pmatrix} u \\ v \\ w \end{pmatrix} = 0, \quad u \in \mathbb{R}^n, v \in \mathbb{R}^m, w \in \mathbb{R}^p, \tag{2.23}$$

which by linear transformations and rearrangement of the equations is equivalent to

$$\bar{K} \begin{pmatrix} u \\ v_J \\ w \end{pmatrix} = 0, \qquad v_i = \frac{s_i}{t_i} \nabla g_i(z)^\top u \ \ \forall i \in I \setminus J.$$

where $v_J = (v_i)_{i \in J}$, given that $t_i \neq 0 \ \forall i \in I \setminus J$ as established by (F2). Hence, if $\bar{K}$ is non-singular, the only solution to the system is $(u, v, w) = (0, 0, 0)$ and if it is singular, (2.23) admits a non-zero solution. □

We can derive sufficient conditions for $V(Z)$ to be non-singular in the same fashion as for the KKT matrix K in Corollary 2.1:

Corollary 2.2. *If the gradients $\{\nabla g_i(z),\ i \in J\} \cup \{\nabla h_j(z),\ j = 1, ..., p\}$ are linearly independent and $\bar{H}$ is positive definite, then matrix $V(Z) \in \partial F(Z)$ is non-singular.*

Proof. $D := \begin{pmatrix} E_J \\ F \end{pmatrix}$ has full rank due to the linear independence, and $C := \bar{H}$ verifies $d^\top \bar{H} d > 0$ for all $d \in \ker(D) \setminus \{0\}$, so the conditions of Lemma 2.1 apply to $\bar{K}$ and from Theorem 2.5, $V(Z)$ is non-singular. □

Note that $\bar{H}$ being positive definite only requires

$$d^\top \bar{H} d = d^\top H d + d^\top diag\left(\left(\frac{s_i}{t_i}\right)_{i \in I \setminus J}\right) E_{I\setminus J}^\top E_{I\setminus J} d > 0$$

for all $d \in \mathbb{R}^n$. From property (F3), $s_i/t_i \geq 0$ for all $i \in J$, so defining

$$e := diag\left(\left(\sqrt{\frac{s_i}{t_i}}\right)_{i \in I \setminus J}\right) E_{I\setminus J} d \in \mathbb{R}^m$$

this condition is equivalent to

$$d^\top H d > -\|e\|_2^2$$

which, in the case that $\|e\|_2^2 > 0$ implies that H does not necessarily have to be positive definite. It is still a requirement on the original Hessian H, but a less restrictive one.

It is again not required that the gradients of all the constraints are linearly independent for the conditions of Corollary 2.2 to apply, but only a subset of them. This allows as well for $V(Z)$ to be non-singular for Problem 2.5 with box constraints, since despite the gradients of all the constraints being present in $V(Z)$, the presence of T allows to eliminate the equations corresponding to the active inequality constraints and transform the system, as done in the proof for Theorem 2.5. Indeed, defining the partition of $J = \{i \in \{1, ..., 2m\} : \tilde{g}_i(z) = 0\}$:

$$J_1 = \{i \in J : g_i(z) = a_i\}, \quad J_2 = \{i \in J : g_i(z) = b_i\},$$

we have that $J_1 \cap J_2 = \emptyset$ since $a_i < b_i$ and if $i \in J$, then either $g_i(z) = a_i$ or $g_i(z) = b_i$, so $J = J_1 \uplus J_2$ and

$$\begin{aligned}&\{\nabla \tilde{g}_i(z), i \in J\} \cup \{\nabla h_j(z),\ j = 1, ..., p\}\\ &= \{-\nabla g_i(z), i \in J_1\} \cup \{\nabla g_i(z), i \in J_2\} \cup \{\nabla h_j(z),\ j = 1, ..., p\}.\end{aligned}$$

can be linearly independent.

Nonetheless, the solvability of system (2.16) cannot always be guaranteed when these conditions are not met, which can pose an issue when trying to find numerical solutions using the nonsmooth Newton method. In these cases, one could try to perform a regularization as in the *modified nonsmooth Newton method* proposed in [72]: it consists in solving the system

$$(V_k^\top V_k + \nu_k I) d = V_k^\top F(Z^k) \tag{2.24}$$

instead of (5.1) for a chosen $\nu_k > 0$. This system is now solvable for every Z and $V(Z) \in \partial F(Z)$. We introduce the main result for the convergence of this modified method from [72]:

Theorem 2.6. *Let Z^* be an accumulation point generated by the modified nonsmooth Newton method version of Algorithm 2.5, and let $\nu_k = \min\{\Theta(Z^k), \left\|\nabla\Theta(Z^k)\right\|\}$. Then Z^* is a stationary point of the merit function $\Theta(Z)$; furthermore, Z^* is a solution of $F(Z) = 0$ and $\{Z^k\}$ converges to Z^* superlinearly if $V^* \in \partial F(Z^*)$ is non-singular.*

Notice that both this theorem and Theorem 2.4 for the original nonsmooth Newton method still require for $V^* \in \partial F(Z^*)$ to be non-singular, and both methods are very similar when $\Theta(Z)$ gets close to 0 and therefore, convergence issues may still arise in the numerical application of the modified method. Furthermore, the advantage of the potentially sparse structure of matrix V_k can be lost when calculating the matrix product in (2.24). In any case, the original nonsmooth Newton method worked well with the heat equation problems treated in Chapter 5 and the fast convergence was not lost.

On a final note, we emphasize that all this development is based on the use of the Fischer-Burmeister function as complementarity function. Some of its properties, such as the differentiability of its square φ^2, are key to the proofs of these results. Other prominent alternatives are the minimum function [36, 101]

$$\varphi(a,b) = \min\{a,b\},$$

or the similar [62, 69]

$$\varphi_c(a,b) = a - \max\{0, a - cb\}$$

for a $c > 0$, whose application leads to the same algorithm of the primal-dual active-set method [62]. There are also parameter-dependent modifications of the Fischer-Burmeister function with similar properties, such as

$$\varphi_c(a,b) = \sqrt{(a-b)^2 + cab} - a - b,$$

for a fixed $c \in (0,4)$ [36]. More choices of complementarity functions are explored in [122].

2.2 General optimal control theory

In this section, we provide an outline of the optimal control theory and some of the most important results. While a great part of the theory focuses on the more general class of DAE optimal control problems, we limit ourselves to optimal control problems involving only ODEs since our goal problems do not involve algebraic equations.

An optimal control problem is formulated in terms of state and control variables. State variables are defined by their dynamic behavior, which can be influenced by the choice of control variables. On top of solving the differential equations that define the dynamics, an objective function is given to be maximized or minimized by the choice of controls. Other elements, such as a choice of static parameters that can influence the objective function or the dynamics, or a set of constraints on states, controls and parameters, can also be added to the problem.

Given the time interval $[t_0, t_f]$, $t_f > t_0$, the general form of an ODE Optimal Control Problem is formulated as follows:

Problem 2.6 (General OCP)**.** *Find a state function $x(\cdot) : [t_0, t_f] \to \mathbb{R}^{n_x}$, a control function $u(\cdot) : [t_0, t_f] \to \mathbb{R}^{n_u}$ and a parameter vector $p \in \mathbb{R}^{n_p}$ that minimize the objective function*

$$\phi(x(t_0), x(t_f), p) + \int_{t_0}^{t_f} f_0(t, x(t), u(t), p)dt \tag{2.25}$$

subject to the differential equation

$$\dot{x}(t) = f(t, x(t), u(t), p) \qquad \forall t \in [t_0, t_f], \tag{2.26}$$

the boundary conditions

$$\psi_0(x(t_0), p) = 0, \qquad \psi_f(x(t_f), p) = 0, \tag{2.27}$$

the mixed control-state constraints

$$c_{min} \leq c(t, x(t), u(t), p) \leq c_{max} \qquad \forall t \in [t_0, t_f], \tag{2.28}$$

and the box constraints

$$\begin{aligned} &u_{min} \leq u(t) \leq u_{max} \qquad \forall t \in [t_0, t_f], \\ &p_{min} \leq p \leq p_{max}, \end{aligned} \tag{2.29}$$

where the functions $\phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}$, $f_0 : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}$, $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$, $c : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_c}$, $\psi_0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_0}$ and $\psi_f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_f}$ are at least twice continuously differentiable. There are numerous transformation techniques to express Problem 2.6 into more general forms. We mention some of them which will be useful in the following development and later in the numerical implementation of the problems:

- Transformation to fixed time interval: Problem 2.6 can be transformed into an equivalent problem with fixed initial and final time with the transformation

  $$t(\tau) := t_0 + \tau(t_f - t_0), \quad \tau \in [0, 1].$$

  Defining as new state and control functions

  $$\begin{aligned} \tilde{x}(\tau) &:= x(t(\tau)), \\ \tilde{u}(\tau) &:= u(t(\tau)), \end{aligned}$$

  the ODE system (2.26) is transformed into

  $$\dot{\tilde{x}}(\tau) = (t_f - t_0) f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)).$$

  In the case that t_0 or t_f are not fixed, in which they are called free initial and final time, respectively, this transformation allows for t_0 and t_f to be included as additional optimization parameters in this formulation.

- Transformation to autonomous problem: by introducing an additional state according to the differential equation

  $$\dot{T}(t) = 1, \quad T(t_0) = t_0,$$

  we can transform a non-autonomous problem into an autonomous one.

- Transformation to a Mayer type problem: by defining an additional state $\tilde{x}(t)$ such that

  $$\dot{\tilde{x}}(t) = f_0(t, (t), u(t), p), \quad \tilde{x}(t_0) = 0,$$

  and $X(t) := (x(t), \tilde{x}(t))$ as the new state function, we reduce the objective function (2.25) to

  $$\tilde{\phi}(X(t_0), X(t_f), p) = \phi(x(t_0), x(t_f), p) + \tilde{x}(t_f).$$

- Transformation of parameters into constant states: for some theoretical results, it is useful to eliminate parameters from the formulation. This can be achieved by defining an additional vector of states $p(t)$ to substitute the parameters such that

  $$\dot{p}(t) = 0$$

  with free $p(t_f)$.

We define as well some function spaces that will be mentioned in what follows:

- $L^\infty([t_0, t_f], \mathbb{R}^m)$ consists of all measurable functions $f := (f_1, ..., f_m) : [t_0, t_f] \rightarrow \mathbb{R}^m$ whose components are essentially bounded, i.e. for all $i = 1, ..., m$

  $$\operatorname*{ess\,sup}_{t \in [t_0, t_f]} |f_i(t)| := \inf \{C \geq 0 : |f_i(t)| \leq C \text{ almost everywhere in } [t_0, t_f]\} < \infty$$

- $W^{1,\infty}([t_0, t_f], \mathbb{R}^m)$ consists of all absolutely continuous functions $f := (f_1, ..., f_m) : [t_0, t_f] \rightarrow \mathbb{R}^m$ whose components hold for all $i = 1, ..., m$:

  $$\|f_i\|_{1,\infty} := \max_{0 \leq j \leq 1} \|f_i^{(j)}\|_\infty < \infty$$

- $BV([t_0, t_f], \mathbb{R}^m)$ is the space of functions of bounded variation $f := (f_1, ..., f_m) : [t_0, t_f] \rightarrow \mathbb{R}^m$ where for every component f_i for $i = 1, ..., m$ there exists a constant $K > 0$ such that for any partition

  $$\mathbb{G}_k := \{t_0 < t_1 < ... < t_k = t_f\}$$

  of $[t_0, t_f]$ it holds

  $$\sum_{j=1}^{k} |f_i(t_j) - f_i(t_{j-1})| \leq K$$

- $NBV([t_0, t_f], \mathbb{R}^m)$ is the space of normalized functions of bounded variations, which consists of all functions $f \in BV([t_0, t_f], \mathbb{R}^m)$ which are continuous from the right in (t_0, t_f) and satisfy $f(t_0) = 0$.

We reformulate Problem 2.6 for the following development with the aforementioned transformations:

Problem 2.7 (OCP)**.** *Minimize*

$$\phi(x(t_0), x(t_f))$$

w.r.t. the state function $x \in W^{1,\infty}([t_0, t_f], \mathbb{R}^{n_x})$ *and the control function* $u \in L^{\infty}([t_0, t_f], \mathbb{R}^{n_u})$*, subject to the constraints*

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), && t \in [t_0, t_f], \\ c(x(t), u(t)) &\leq 0, && t \in [t_0, t_f], \\ \psi_0(x(t_0)) &= 0, \\ \psi_f(x(t_f)) &= 0. \end{aligned}$$

We formulate now the local minimum principle for ODE optimal control problems with mixed control-state constraints. We distinguish between pure state constraints and mixed control-state constraints, i.e. $c(x(t), u(t)) = (c_s(x(t)), c_u(x(t), u(t)))$ with $c_s : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{c_s}}$, $c_u : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{c_u}}$. The augmented Hamilton function $\hat{\mathcal{H}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_{c_u}} \to \mathbb{R}$ is defined as

$$\widehat{\mathcal{H}}(x, u, \lambda, \lambda_u) := \lambda^\top f(x, u) + \lambda_u^\top c_u(x, u).$$

According to [41, Chapter 3], under certain assumptions on functions $J, \phi,\ c, \psi_0, \psi_f$ and a constraint qualification, and given a local minimum $(\hat{x}, \hat{u})$ of (OCP), there exist multipliers $\kappa \in \mathbb{R}$, $\sigma_0 \in \mathbb{R}^{n_0}, \sigma_f \in \mathbb{R}^{n_f}$, $\lambda \in BV([t_0, t_f], \mathbb{R}^{n_x})$, $\lambda_u \in L^{\infty}([t_0, t_f], \mathbb{R}^{n_u})$ and $\lambda_s \in NBV([t_0, t_f], \mathbb{R}^{n_{c_s}})$ such that the following conditions are satisfied:

Theorem 2.7 (Local minimum principle)**.** *Let the following assumptions hold for Problem 2.7:*

- *Assumptions on the smoothness of functions* ϕ, f, c_u, c_s, ψ *hold (see [41, Chapter 3]).*
- $(\hat{x}, \hat{u})$ *is a local minimum of Problem 2.7.*
- *For* $\gamma(t) := (c_u)'_u(\hat{x}(t), \hat{u}(t))^\top$ *it holds* $rank(\gamma(t)) = n_{c_u}$ *almost everywhere in* $[t_0, t_f]$ *and its pseudo-inverse*

  $$(\gamma(t))^+ = \gamma(t)^\top (\gamma(t)\gamma(t)^\top)^{-1}$$

  is essentially bounded.

Then there exist multipliers

$$\kappa \in \mathbb{R}, \quad \lambda \in BV([t_0, t_f], \mathbb{R}^{n_x}), \quad \lambda_u \in L^\infty([t_0, t_f], \mathbb{R}^{n_u}),$$
$$\lambda_s \in NBV([t_0, t_f], \mathbb{R}^{n_{c_s}}), \quad \sigma \in \mathbb{R}^{n_\psi},$$

such that the following conditions are satisfied:

(a) $\kappa \geq 0, \ (\kappa, \lambda, \lambda_u, \lambda_s, \sigma_0, \sigma_f) \neq 0.$

(b) Adjoint equations: Almost everywhere in $[t_0, t_f]$ *it holds*

$$\begin{aligned}\lambda(t) = \lambda(t_f) + \int_t^{t_f} \nabla_x \widehat{\mathcal{H}}(\hat{x}(\tau), \hat{u}(\tau), \lambda(\tau), \lambda_u(\tau)) \ d\tau \\ + \int_t^{t_f} \nabla_x c_s(\hat{x}(\tau)) \ d\lambda_s(\tau)\end{aligned}$$

(c) Transversality conditions:

$$\begin{aligned}\lambda(t_0)^\top &= -(\kappa \phi'_{x_0}(\hat{x}(t_0), \hat{x}(t_f)) + \sigma_0^\top \psi'_0(\hat{x}(t_0))) \\ \lambda(t_f)^\top &= \kappa \phi'_{x_f}(\hat{x}(t_0), \hat{x}(t_f)) + \sigma_f^\top \psi'_f(\hat{x}(t_f))\end{aligned}$$

(d) Optimality conditions: Almost everywhere in $[t_0, t_f]$ *it holds*

$$\nabla_u \widehat{\mathcal{H}}(\hat{x}(t), \hat{u}(t), \lambda(t), \lambda_u(t)) = 0$$

(e) Complementarity conditions: Almost everywhere in $[t_0, t_f]$ *it holds*

$$\lambda_u(t)^\top c_u(\hat{x}(t), \hat{u}(t)) = 0 \text{ and } \lambda_u(t) \geq 0,$$

$\lambda_{s,i}, i \in \{1, ..., n_{c_s}\}$, *is monotonically increasing in* $[t_0, t_f]$ *and constant in every interval* (t_1, t_2) *with* $t_1 < t_2$ *and* $c_{s,i}(\hat{x}(t)) < 0$ *for all* $t \in (t_1, t_2)$.

These necessary conditions are not only important from a theoretical point of view; they provide the basis for an indirect approach to solving optimal control problems numerically as well. There is also a relationship between this continuous version of the necessary conditions for infinite dimensional optimal control problems and the application of necessary optimality conditions to their finite dimensional discretization, which will be exposed later.

There are many approaches one can take to solve (OCP). A first distinction can be made between discretization methods and function space methods. Discretization methods follow a *first discretize, then optimize* approach, according to which the problem is approximated by a finite dimensional optimization problem using suitable discretization and integration schemes. On the other hand, the *first optimize, then discretize* approach or function space approach considers the optimal control problem as an infinite dimensional optimization problem. As our interest is to find numerical solutions, we only consider the first approach in the frame of this work.

Another distinction can be made between direct and indirect methods. The indirect approach is based on deriving a Boundary Value Problem (BVP) from the necessary optimality conditions that draw from the local minimum principle. This leads to very accurate numerical solutions, but requires a very good initial approximate solution, and it often leads to cumbersome calculations to solve the optimality system, which defies our purpose of finding efficient strategies to solve large-scale, complex problems numerically. On the other hand, direct methods use a suitable discretization to obtain a finite dimensional optimization problem that can be solved by suitable optimization methods. Therefore, we focus on direct discretization approaches and the different discretization, integration and optimization methods that can be employed to implement them.

2.3 Direct discretization approach

We consider the time grid with constant step $h = (t_f - t_0)/N$:

$$\mathbb{G}_N := \{t_i = t_0 + ih,\ i = 0, ..., N\}. \tag{2.30}$$

We start by defining a control discretization. Given a control function vector $u(\cdot) \in L^\infty([t_0, t_f], \mathbb{R}^{n_u})$, we consider a basis of B-splines of order k, $\{B_i^k(\cdot), i = 1, ..., N+k-1\}$. We define a control parametrization by choosing a vector of control parameters $w \in \mathbb{R}^M$, where $M := n_u(N + k - 1)$, that defines

$$u(\cdot) \approx \sum_{i=1}^{N+k-1} w_i B_i^k(\cdot), \tag{2.31}$$

where $w_i \in \mathbb{R}^{n_u},\ i = 1, ..., N + k - 1$. B-splines are defined as follows: Let $k \in \mathbb{N}$ and $\mathbb{G}_N$ as in (2.30). We define the auxiliary grid

$$\mathbb{G}_N^k := \{\tau_i,\ i = 1, ..., N + 2k - 1\} \tag{2.32}$$

with auxiliary grid points

$$\tau_i := \begin{cases} t_0, & \text{if } 1 \leq i \leq k, \\ t_{i-k}, & \text{if } k + 1 \leq i \leq N + k - 1, \\ t_N, & \text{if } N + k \leq i \leq N + 2k - 1. \end{cases}$$

The elementary B-splines $B_i^k(\cdot)$ of order $k,\ i = 1, ..., N + k - 1$, are defined as

$$B_i^1(t) := \begin{cases} 1, & \text{if } \tau_i \leq t \leq \tau_{i+1}, \\ 0, & \text{otherwise}, \end{cases}$$
$$B_i^k(t) := \frac{t - \tau_i}{\tau_{i+k-1} - \tau_i} B_i^{k-1}(t) + \frac{\tau_{i+k} - t}{\tau_{i+k} - \tau_{i+1}} B_{i+1}^{k-1}(t).$$

With this methodology, control functions are represented as a function of parameters $w_i,\ i = 1, ..., N + k - 1$. Choosing $k = 1$ or $k = 2$, a piecewise constant or linear approximation is obtained, respectively.

In order to discretize the states, an ODE discretization method for the initial value problem

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0$$

is required. Approximating the states by their values at the temporal nodes in $\mathbb{G}_N$ as $x(t_i) \approx x_i$ and the controls by their B-spline approximations $u(t) \approx u(t; w)$ for a certain $w \in \mathbb{R}^M$, we focus on methods with the general form

$$x_{i+1} = x_i + h\Phi(t_i, x_i, w, h), \quad i = 0, ..., N - 1 \tag{2.33}$$

being Φ an appropriate increment function for so-called one-step methods. The wide variety of Runge-Kutta methods are included in this formulation: For a given $s \in \mathbb{N}$ and coefficients $b_j, c_j, a_{ij},\ i, j = 1, ..., s$, the s-stage Runge-Kutta method is defined by

$$x_{i+1} = x_i + h \sum_{j=1}^{s} b_j k_j(t_i, x_i, w, h),$$

where the stage derivatives $k_j,\ j = 1, ..., s$, are defined by

$$k_j(t_i, x_i, w, h) = f(t_i + c_j h, x_i + h \sum_{l=1}^{s} a_{jl} k_l(t_i, x_i, w, h), u(t_i + c_j h; w)).$$

Note that we can obtain implicit methods from this formulation: For example, with

$$\Phi(t_i, x_i, w, h) = \frac{1}{2} f(t_i, x_i, u(t_i; w)) + \frac{1}{2} f(t_i + h, x_i + h\Phi(t_i, x_i, w, h), u(t_i + h; w)),$$

we obtain the implicit trapezoidal rule

$$x_{i+1} = x_i + \frac{h}{2} \left(f(t_i, x_i, u(t_i; w)) + f(t_{i+1}, x_{i+1}, u(t_{i+1}; w)) \right). \tag{2.34}$$

We distinguish now between the full discretization approach and the reduced discretization approach. The key difference between them that will be exploited in the development of this thesis is that the first leads to a large, sparse nonlinear optimization problem, whereas the second leads to a small, dense one. The different methodologies and possibilities that this distinction implicates will be discussed thoroughly in the next chapters.

2.3.1 Full discretization

With the discretized states and controls, we obtain directly from (OCP) the fully discretized problem (FDP) by discretizing the constraints on the grid $\mathbb{G}_N$ and substuting the ODE by its discretized version with a one-step method:

Problem 2.8 (FDP). *Minimize*

$$\phi(x_0, x_N)$$

with respect to $x \in \mathbb{R}^{(N+1)\cdot n_x}$ *and* $w \in \mathbb{R}^M$ *subject to the constraints*

$$\begin{aligned} x_{i+1} - x_i - h\Phi(t_i, x_i, w, h) &= 0, & i &= 0, ..., N-1, \\ c(x_i, u(t_i; w)) &\le 0, & i &= 0, ..., N, \\ \psi_0(x_0) &= 0, \\ \psi_N(x_N) &= 0. \end{aligned}$$

Note that this is a discretized version of the simplified (OCP) where a transformation to a Mayer problem has been applied, but in the case that an objective function features the integral of a function $f_0(x(t), u(t))$ as in Problem 2.6, we can apply a numerical integration method to approximate the integral instead of using this transformation to eliminate it. For example, with the trapezoidal rule for integration, we would just add to the objective function the sum

$$\frac{h}{2}(f_0(x_0, u(t_0; w)) + f_0(x_N, u(t_N; w))) + h\sum_{i=1}^{N-1} f_0(x_i, u(t_i; w)). \tag{2.35}$$

We now expose the relation between the necessary conditions formulated in the local minimum principle (2.7) for (OCP) and the formulation of the necessary KKT conditions (2.5)-(2.7) for the fully discretized problem (FDP). We distinguish again between pure state constraints and mixed control-state constraints, i.e. $c(x, u) = (c_s(x), c_u(x, u))$ with $c_s : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{c_s}}$, $c_u : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{c_u}}$. The discrete Hamilton function is defined as

$$\widehat{\mathcal{H}}_h(t, x, w, \lambda, \lambda_u, h) = \lambda^\top \Phi(t, x, w, h) + \lambda_u^\top c_u(x, u(t; w)).$$

Applying first order necessary optimality conditions for finite optimization problems to (FDP) and again assuming a constraint qualification, we adapt the discrete local minimum principle [41, Chapter 5] to the case of our general ODE integration scheme (2.33). Given a local minimum $(\hat{x}, \hat{w})$ of (FDP), there exist multipliers $\sigma_0 \in \mathbb{R}^{n_0}, \sigma_f \in \mathbb{R}^{n_f}$, $\lambda = (\lambda_0, ..., \lambda_N)^\top$, $\lambda_u = (\lambda_{u,0}, ..., \lambda_{u,N})^\top$, and $\lambda_s = (\lambda_{s,0}, ..., \lambda_{s,N})^\top$ such that the following conditions are satisfied:

(a) $(\lambda, \lambda_u, \lambda_s, \sigma_0, \sigma_f) \neq 0$.

(b) *Discrete adjoint equations*: For $i = 0, ..., N-1$ it holds

$$\begin{aligned}\lambda_i &= \lambda_{i+1} + h\nabla_x \widehat{\mathcal{H}}_h(t_i, \hat{x}_i, \hat{w}, \lambda_{i+1}, \lambda_{u,i}, h) + \nabla_x c_s(\hat{x}_i)\lambda_{s,i} \\ &= \lambda_N + \sum_{j=i}^{N-1} h\nabla_x \widehat{\mathcal{H}}_h(t_j, \hat{x}_j, \hat{w}, \lambda_{j+1}, \lambda_{u,j}, h) + \sum_{j=i}^{N-1} \nabla_x c_s(\hat{x}_j)\lambda_{s,j}.\end{aligned}$$

(c) *Discrete transversality conditions*:

$$\begin{aligned}\lambda_0^\top &= -(\phi'_{x_0}(\hat{x}_0, \hat{x}_N) + \sigma_0^\top \psi'_0(\hat{x}_0)), \\ \lambda_N^\top &= \phi'_{x_N}(\hat{x}_0, \hat{x}_N) + \sigma_N^\top \psi'_f(\hat{x}_N) + \lambda_{s,N}^\top c'_s(\hat{x}_N).\end{aligned}$$

(d) *Discrete optimality conditions*:

$$\nabla_w \sum_{i=0}^{N-1} \widehat{\mathcal{H}}_h(t_i, \hat{x}_i, \hat{w}, \lambda_i, \lambda_{u,i}, h) = 0.$$

(e) *Complementarity conditions*:

$$\begin{aligned} &\lambda_{u,i} \geq 0, \quad \lambda_{u,i}^\top c_u(\hat{x}_i, u(t_i; \hat{w})) = 0 && i = 0, ..., N-1, \\ &\lambda_{s,i} \geq 0, \quad \lambda_{s,i}^\top c_s(\hat{x}_i) = 0 && i = 0, ..., N. \end{aligned}$$

Proof. Given the Lagrangian of (FDP)

$$\begin{aligned} L(x, w, \tilde{\lambda}, \tilde{\lambda}_u, \tilde{\lambda}_s, \sigma_0, \sigma_f) &= \phi(x_0, x_N) + \sigma_0^\top \psi_0(x_0) + \sigma_f^\top \psi_f(x_N) \\ &+ \sum_{i=0}^{N-1} \widehat{\mathcal{H}}_h(t_i, x_i, w, \tilde{\lambda}_i, \tilde{\lambda}_{u,i}, h) + \sum_{i=0}^{N-1} \tilde{\lambda}_i^\top \frac{x_i - x_{i+1}}{h} + \sum_{i=0}^{N} \tilde{\lambda}_{s,i}^\top c_s(x_i), \end{aligned}$$

application of the first order necessary optimality conditions for the local minimum $(\hat{x}, \hat{w})$ results in the equations:

1. $\nabla_w L = 0$ for all $i = 0, ..., N-1$, therefore it holds:

$$\nabla_w \sum_{i=0}^{N-1} \widehat{\mathcal{H}}_h(t_i, \hat{x}_i, \hat{w}, \tilde{\lambda}_i, \tilde{\lambda}_{u,i}, h) = 0,$$

2. $\nabla_{x_i} L = 0$, therefore it holds for all $i = 1, .., N-1$

$$\frac{1}{h}\tilde{\lambda}_i - \frac{1}{h}\tilde{\lambda}_{i-1} + \nabla_x \widehat{\mathcal{H}}_h(t_i, \hat{x}_i, \hat{w}, \tilde{\lambda}_i, \tilde{\lambda}_{u,i}, h) + \nabla_x c_s(\hat{x}_i)\tilde{\lambda}_{s,i} = 0,$$

for $i = 0$

$$\begin{aligned} &\frac{1}{h}\tilde{\lambda}_0 + \nabla_x \widehat{\mathcal{H}}_h(t_0, \hat{x}_0, \hat{w}, \tilde{\lambda}_0, \tilde{\lambda}_{u,0}, h) \\ &\quad + (\phi'_{x_0}(\hat{x}_0, \hat{x}_N) + \sigma_0^\top \psi'_0(\hat{x}_0))^\top + \nabla_x c_s(\hat{x}_0)\tilde{\lambda}_{s,0} = 0 \end{aligned}$$

and for $i = N$

$$-\frac{1}{h}\tilde{\lambda}_{N-1} + (\phi'_{x_N}(\hat{x}_0, \hat{x}_N) + \sigma_f^\top \psi'_f(\hat{x}_N))^\top + \nabla_x c_s(\hat{x}_N)\tilde{\lambda}_{s,N} = 0.$$

Defining

$$\begin{aligned}\lambda_i &:= \frac{1}{h}\tilde{\lambda}_{i-1}, \quad i = 1, ..., N,\\ \lambda_0 &:= -(\phi'_{x_0}(\hat{x}_0, \hat{x}_N) + \sigma_0^\top \psi_0'(\hat{x}_0))^\top,\\ \lambda_{u,i} &:= \frac{1}{h}\tilde{\lambda}_{u,i}, \quad i = 1, ..., N,\end{aligned}$$

we obtain the previously described conditions (a)-(e). □

We can interpret these conditions as discrete versions of the continuous case exposed above with the following interpretation of the variables and multipliers:

$$\hat{x_i} \approx \hat{x}(t_i), \quad u(t_i; \hat{w}) \approx \hat{u}(t_i), \quad \lambda_i \approx \lambda(t_i), \quad \lambda_{u,i} \approx \lambda_u(t_i), \quad \lambda_{s,i} \approx \lambda_s(t_{i+1}) - \lambda_s(t_i),$$

following a similar development to [41, Section 5.4], only observing that the sum in the discrete adjoint conditions involving the discrete Hamilton function can be recognized as a Riemann sum on $\mathbb{G}_N$, and thus

$$\sum_{j=i}^{N-1} h\nabla_x \widehat{\mathcal{H}}_h(t_j, \hat{x}_j, \hat{w}, \lambda_{j+1}, \lambda_{u,j}, h) \approx \int_{t_i}^{t_f} \nabla_x \widehat{\mathcal{H}}(\hat{x}(\tau), \hat{u}(t), \lambda(\tau), \lambda_u(\tau))d\tau,$$

given that

$$\int_{t_i}^{t_{i+1}} f(\hat{x}(t), \hat{u}(t))dt = \hat{x}(t_{i+1}) - \hat{x}(t_i) \ \approx \ \hat{x}_{i+1} - \hat{x}_i = h\Phi(t_i, \hat{x}_i, \hat{w}, h).$$

The derivation of the discrete Hamilton function will also require calculating $\Phi'_x(t, x, w, h)$, which is not always trivial. We expose the relation in the case of the implicit trapezoidal rule and considering a piecewise-linear approximation for the controls, which means that $u(t_i; w) = w_i$ and the control is defined by its values at the nodes $w = (u_0, ..., u_N)$:

$$x_{i+1} = x_i + \frac{h}{2}(f(x_i, u_i) + f(x_{i+1}, u_{i+1})), \quad i = 0, ..., N-1.$$

Leaving out state and mixed control-state constraints, we would obtain as discrete adjoint equations for $i = 0, ..., N-1$:

$$\begin{aligned}\lambda_i^\top &= \lambda_{i+1}^\top + \frac{h}{2}(\lambda_{i+1}^\top f'_x(\hat{x}_i, \hat{u}_i) + \lambda_i^\top f'_x(\hat{x}_{i+1}, \hat{u}_{i+1}))\\ &\approx \lambda(t_{i+1})^\top + \int_{t_i}^{t_{i+1}} \lambda(\tau)^\top f'_x(\hat{x}(\tau), \hat{u}(\tau))d\tau.\end{aligned}$$

Regarding the state constraints, we can interpret the sum in (a) as

$$\sum_{j=i}^{N-1} c_s'(\hat{x}(t_j))^\top (\lambda_s(t_{j+1}) - \lambda_s(t_j)) \approx \int_{t_i}^{t_N} \nabla_x c_s(\hat{x}(\tau)) d\lambda_s(\tau).$$

For $i = N$, the discrete multipliers can be interpreted as

$$\begin{aligned} \lambda_{s,N} &\approx \lambda_s(t_f) - \lambda_s(t_f^-), \\ \lambda_N &\approx \lambda(t_f^-), \end{aligned}$$

given that λ_s and λ can jump at t_f (see [41] for more details).

While it might seem like a disadvantage in the case of full discretization to require a nonlinear optimization software to solve a large problem in every iteration, the sparsity allows for the structure to be exploited or even for the problem to be reduced into smaller subproblems. By considering again a piecewise-linear approximation to the control and expressing as the optimization variable

$$z := (x_0, u_0, x_1, u_1, ..., x_N, u_N)^\top \in \mathbb{R}^{(nx+nu)(N+1)},$$

and as functions, using the trapezoidal rule (2.34) for ODE discretization,

$$f(z) := \phi(x_0, x_N) + \frac{h}{2}(f_0(x_0, u_0) + f_0(x_N, u_N)) + h \sum_{i=1}^{N-1} f_0(x_i, u_i) \tag{2.36}$$

$$g(z) := \begin{pmatrix} c(x_0, u_0) \\ c(x_1, u_1) \\ \vdots \\ c(x_N, u_N) \end{pmatrix} \tag{2.37}$$

$$h(z) := \begin{pmatrix} x_1 - x_0 - \frac{h}{2}(f(x_0, u_0) + f(x_1, u_1)) \\ \vdots \\ x_N - x_{N-1} - \frac{h}{2}(f(x_{N-1}, u_{N-1}) + f(x_N, u_N)) \\ \psi_0(x_0) \\ \psi_f(x_N) \end{pmatrix}, \tag{2.38}$$

(FDP) can be considered as a general nonlinear optimization problem in the form of (NLP) and therefore, the nonlinear optimization methods described in Section 2.1 can be applied to solve it. However, due to the large-scale nature of this type of problems, the sparsity of the derivatives must be exploited by the solver in order to reduce the computational effort and be able to find a solution in an acceptable computational time. A structure-exploiting strategy for a nonsmooth Newton method and its application to fully discretized, large-scale problems is presented in Chapter 5. A study on the non-singularity of the KKT matrix (2.10) of (FDP) under certain conditions can be found in [89].

2.3.2 Reduced discretization

The reduced discretization approach differs from the full discretization in that the equations resulting from the discretization of the ODE are not part of the optimization step, but instead solved recursively. Given that every x_{i+1} is completely defined by the initial value x_0, the control parametrization w and the one-step method (2.33) defined by Φ, we can solve the following equations recursively and obtain the values of the states in all the time nodes:

$$\begin{aligned}
X_0(x_0, w) &:= x_0, && (2.39)\\
X_1(x_0, w) &:= X_0(x_0, w) + h\Phi(t_0, X_0(x_0, w), w, h), && (2.40)\\
X_2(x_0, w) &:= X_1(x_0, w) + h\Phi(t_1, X_1(x_0, w), w, h), && (2.41)\\
&\vdots && (2.42)\\
X_N(x_0, w) &:= X_{N-1}(x_0, w) + h\Phi(t_{N-1}, X_{N-1}(x_0, w), w, h). && (2.43)
\end{aligned}$$

We obtain this way the Reduced Discretization Problem (RDP):

Problem 2.9 (RDP). *Minimize*

$$\phi(X_0(x_0, w), X_N(x_0, w))$$

with respect to $x_0 \in \mathbb{R}^{n_x}$ *and* $w \in \mathbb{R}^M$ *subject to the constraints*

$$\begin{aligned}
c(X_i(x_0, w), u(t_i; w)) &\leq 0, && i = 0, ..., N,\\
\psi_0(X_0(x_0, w)) &= 0,\\
\psi_N(X_N(x_0, w)) &= 0.
\end{aligned}$$

(RDP) is again a finite dimensional nonlinear optimization problem with the form of (NLP) with

$$z := (x_0, w) \in \mathbb{R}^{n_x+M}$$

as optimization variable, and defined by the functions

$$f(z) := \phi(X_0(x_0, w), X_N(x_0, w)), \tag{2.44}$$

$$g(z) := \begin{pmatrix} c(X_0(x_0, w), u(t_0; w)) \\ c(X_1(x_0, w), u(t_1; w)) \\ \vdots \\ c(X_N(x_0, w), u(t_N; w)) \end{pmatrix} \tag{2.45}$$

$$h(z) := \begin{pmatrix} \psi_0(X_0(x_0, w)) \\ \psi_f(X_N(x_0, w)) \end{pmatrix}. \tag{2.46}$$

The size of the variable of (RDP) is much smaller than (FDP), but the derivatives are not sparse anymore and must be calculated by the chain rule:

$$f'(z) = (\phi'_{x_0} \cdot X'_{0,x_0} + \phi'_{x_N} \cdot X'_{N,x_0} \mid \phi'_{x_0} \cdot X'_{0,w} + \phi'_{x_N} \cdot X'_{N,w}),$$

$$g'(z) = \left(\begin{array}{c|c} c'_x(z_0) \cdot X'_{0,x_0} & c'_x(z_0) \cdot X'_{0,w} + c'_u(z_0) \cdot u'_w(t_0; w) \\ c'_x(z_1) \cdot X'_{1,x_0} & c'_x(z_1) \cdot X'_{1,w} + c'_u(z_1) \cdot u'_w(t_0; w) \\ \vdots & \vdots \\ c'_x(z_N) \cdot X'_{N,x_0} & c'_x(z_N) \cdot X'_{N,w} + c'_u(z_N) \cdot u'_w(t_0; w) \end{array} \right)$$

$$h'(z) = \left(\begin{array}{c|c} \psi'_{0,x}(X_0) \cdot X'_{0,x_0} & \psi'_{0,x}(X_0) \cdot X'_{0,w} \\ \psi'_{f,x}(X_N) \cdot X'_{N,x_0} & \psi'_{f,x}(X_N) \cdot X'_{N,w} \end{array} \right),$$

where $z_i := (X_i(x_0, w), u(t_i; w))$ and abbreviating $X_i := X_i(x_0, w)$. The sensitivities

$$S_i := X'_i(x_0, w), \quad i = 0, ..., N,$$

are calculated taking advantage of the following relationships obtained from the differentiation of system (2.39)-(2.43):

$$S_{i+1} = S_i + h \left(\Phi'_z(t_i, X_i(x_0, w), w, h) \cdot S_i + \Phi'_w(t_i, X_i(x_0, w), w, h) \cdot \frac{\partial w}{\partial z} \right) \tag{2.47}$$

for $i = 0, ..., N-1$ and with $\frac{\partial w}{\partial z} := (0_{\mathbb{R}^{Mn_z}} | I_M)$. Note that computing these derivatives is reduced to solving one initial value problem, and the size of the sensitivity equations (2.47) to be solved depends on the number of unknowns (x_0, w) but not on the constraints of the optimization problem. For more details on how to compute the sensitivities and the derivatives Φ'_z and Φ'_w, which can be trivial (case of the explicit Euler method) or not depending on the chosen one-step method, consult [41, Section 5.3] .

2.3.3 PDE discretization with the method of lines

We focus now on the one dimensional heat or diffusion equation for this introduction to PDE discretization in the context of optimal control problems:

$$\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}, \qquad (t, x) \in (0, t_f) \times (0, D) \tag{2.48}$$

where $T = T(t, x)$ is the temperature as a function of position x along a line and time t, and $k > 0$ is the thermal diffusivity. To solve (2.48), initial conditions at $t = 0$ and boundary conditions at $x = 0$ and $x = D$ are required. For the initial condition, a distribution of the initial temperature $T(0, x) = f_0(x)$ is needed. As for boundary conditions, there are several possibilities:

- Dirichlet conditions: temperature distributions at $x = 0$ and $x = L$ are specified as

$$T(t, 0) = a_0(t), \qquad T(t, D) = a_1(t). \tag{2.49}$$

- Neumann conditions: the spatial derivatives of temperature distributions at $x = 0$ and $x = L$ are specified as

$$\frac{\partial T}{\partial x}(t, 0) = b_0(t), \qquad \frac{\partial T}{\partial x}(t, D) = b_1(t). \tag{2.50}$$

- Mixed boundary conditions: a combination of the prior.

Diverse methods are available to discretize and solve PDEs such as the heat equation. However, we find ourselves already having effective methods that solve ODEs in the context of optimal control problems and therefore, a full discretization is not needed to solve it numerically, so we limit ourselves to discretizing the equation along the spatial variables using the so-called *method of lines*.

The philosophy behind the method of lines is to replace the spatial derivatives in the PDE with algebraic finite difference approximations in terms of the independent time variable, reducing it to a system of ODEs that approximate the original PDE. Once this is done, we can use any integration algorithm for initial value problems to obtain a numerical solution to the PDE. It is particularly useful when the use of existing, well established numerical methods for ODEs is desired. For results on convergence, stability and accuracy of the method of lines, the reader is referred to [116, 132, 141]. When using Runge-Kutta methods to integrate the ODE system, the order of convergence of the method of lines can be reduced [115, 131]; however, this only takes place in the case of Runge-Kutta methods with an order greater or equal to 3, and the negative effects are not likely to be important in practice [114].

We consider $N \in \mathbb{N}$ as the size of the spatial grid, and we define $\delta := D/N$ as the grid step. Let the grid points be

$$x_i = i\delta, \quad i = 0, 1, ..., N.$$

From Taylor's Theorem, we can derive the second order central difference

$$\frac{\partial^2 T}{\partial x^2}(t, x_i) \approx \frac{1}{\delta^2}(T(t, x_{i-1}) - 2T(t, x_i) + T(t, x_{i+1})). \tag{2.51}$$

Approximating $T(t, x_i) \approx T_i(t)$, we obtain a collection of functions $\{T_0(t), T_1(t), ..., T_N(t)\}$ that are determined by the system of ODEs

$$\frac{\partial T_i}{\partial t}(t) = \frac{k}{\delta^2}\left(T_{i-1}(t) - 2T_i(t) + T_{i+1}(t)\right), \quad i = 1, ..., N-1, \tag{2.52}$$

in the case of the internal nodes. Functions at the boundary nodes $T_0(t)$ and $T_N(t)$ are defined, in the case of Dirichlet boundary conditions, as

$$T_0(t) = a_0(t), \qquad T_N(t) = a_1(t).$$

In the case of Neumann boundary conditions as in (2.50), the first spatial derivative can be approximated again from Taylor's theorem as with the central differences

$$\frac{\partial T}{\partial x}(t, x_0) \approx \frac{T(t, x_0 + \delta) - T(t, x_0 - \delta)}{2\delta}, \tag{2.53}$$

$$\frac{\partial T}{\partial x}(t, x_N) \approx \frac{T(t, x_N + \delta) - T(t, x_N - \delta)}{2\delta}. \tag{2.54}$$

Defining the functions $T_{-1}(t)$ and T_{N+1} at the fictitious nodes x_{-1}, x_{N+1} outside of the spatial grid, we obtain

$$\begin{aligned}
\frac{\partial T}{\partial x}(t, x_0) = b_0(t) \approx \frac{T_1(t) - T_{-1}(t)}{2\delta} \quad &\Rightarrow \; T_{-1}(t) = T_1(t) - 2\delta b_0(t)\\
\frac{\partial T}{\partial x}(t, x_N) = b_1(t) \approx \frac{T_{N+1}(t) - T_N(t)}{2\delta} \quad &\Rightarrow \; T_{N+1}(t) = 2\delta b_1(t) + T_N(t)
\end{aligned}$$

and use these vales to extend (2.52) to $i = 0, N$ substituting $T_{-1}(t)$ and $T_{N+1}(t)$ with these expressions:

$$\frac{\partial T_0}{\partial t}(t) = \frac{k}{\delta^2}\left(2T_1(t) - 2T_0(t)\right) - \frac{2k}{\delta} b_0(t), \tag{2.55}$$

$$\frac{\partial T_N}{\partial t}(t) = \frac{k}{\delta^2}\left(2T_{N-1}(t) - 2T_N(t)\right) + \frac{2k}{\delta} b_1(t). \tag{2.56}$$

As initial values for each of the $T_i(t)$ functions, we use the original PDE initial condition $T(0, x) = f_0(x)$ and define

$$T_i(0) = f_0(x_i), \quad i = 0, 1, ..., N.$$

Considering Neumann boundary conditions, we can write the discretized ODE system obtained from (2.52), (2.55) and (2.56) as

$$\begin{pmatrix} \frac{\partial T_0}{\partial t}(t) \\ \frac{\partial T_1}{\partial t}(t) \\ \vdots \\ \frac{\partial T_N}{\partial t}(t) \end{pmatrix} = \frac{k}{\delta^2} \begin{pmatrix} -2 & 2 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 1 & -2 & 1 \\ 0 & \cdots & 0 & 2 & -2 \end{pmatrix} \cdot \begin{pmatrix} T_0(t) \\ T_1(t) \\ \vdots \\ T_N(t) \end{pmatrix} + \frac{2k}{\delta} \begin{pmatrix} -b_0(t) \\ 0 \\ \vdots \\ 0 \\ b_1(t) \end{pmatrix}, \tag{2.57}$$

which exhibits a sparse and diagonal pattern due to the method of lines discretization.

We now extend this methodology to the case of a two-dimensional heat equation:

$$\frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) \qquad (t, x, y) \in (0, t_f) \times (0, D_x) \times (0, D_y),$$

where $T = T(t, x, y)$ is the temperature as a function of position (x, y) along a rectangular surface and time t. Considering the step sizes $\delta_x := D_x/N$, $\delta_y := D_y/M$ with $N, M \in \mathbb{N}$, the grid points are defined as

$$(x_i, y_j) = (i\delta_x, j\delta_y), \quad i = 0, 1, ..., N, \; j = 0, 1, ..., M.$$

Approximating again $T(t, x_i, y_j) \approx T_{ij}(t)$, we can obtain applying (2.51) analogously for both dimensions:

$$\frac{\partial T_{ij}}{\partial t}(t) = \frac{k}{\delta_x^2}\left(T_{i-1,j}(t) - 2T_{ij}(t) + T_{i+1,j}(t)\right) + \frac{k}{\delta_y^2}\left(T_{i,j-1}(t) - 2T_{ij}(t) + T_{i,j+1}(t)\right),$$
$$i = 1, ..., N-1, \; j = 1, ..., M-1. \tag{2.58}$$

Neumann boundary conditions

$$\frac{\partial T}{\partial x}(t, 0, y) = c_0(t, y), \qquad \frac{\partial T}{\partial x}(t, D_x, y) = c_1(t, y), \tag{2.59}$$
$$\frac{\partial T}{\partial y}(t, x, 0) = d_0(t, x), \qquad \frac{\partial T}{\partial y}(t, x, D_y) = d_1(t, x), \tag{2.60}$$

are used as in (2.53)-(2.56) to obtain the temperatures at the fictitious nodes through central difference expressions and with that, extend (2.58) to the boundary nodes: for example, when $i = 0$ we obtain

$$\begin{aligned}
\frac{\partial T_{00}}{\partial t}(t) &= \frac{k}{\delta_x}\left(\frac{2T_{10}(t) - 2T_{00}(t)}{\delta_x} - 2c_0(t, 0)\right) \\
&\quad + \frac{k}{\delta_y}\left(\frac{2T_{01}(t) - 2T_{00}(t)}{\delta_y} - 2d_0(t, 0)\right), \\
\frac{\partial T_{0j}}{\partial t}(t) &= \frac{k}{\delta_x}\left(\frac{2T_{1j}(t) - 2T_{0j}(t)}{\delta_x} - 2c_0(t, y_j)\right) \\
&\quad + \frac{k}{\delta_y^2}\left(T_{0,j-1}(t) - 2T_{0j}(t) + T_{0,j+1}(t)\right), \qquad j = 1, ..., M-1, \\
\frac{\partial T_{0M}}{\partial t}(t) &= \frac{k}{\delta_x}\left(\frac{2T_{1M}(t) - 2T_{0M}(t)}{\delta_x} - 2c_0(t, D_y)\right) \\
&\quad + \frac{k}{\delta_y}\left(2d_1(t, 0) - \frac{2T_{0M}(t) - 2T_{0,M-1}(t)}{\delta_y}\right)
\end{aligned}$$

and analogously, we can obtain the partial time derivatives of $T_{Nj}(t)$, $T_{i0}(t)$ and $T_{iM}(t)$ for $i = 1, ..., N, \; j = 1, ..., M-1$.

Defining the submatrices

$$A := \begin{pmatrix} a & 2b & 0 & \cdots & 0 \\ b & a & b & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & b & a & b \\ 0 & \cdots & 0 & 2b & a \end{pmatrix}, \quad B := \begin{pmatrix} c & & \\ & \ddots & \\ & & c \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)},$$

where $a := -2k/\delta_x^2 - 2k/\delta_y^2$, $b := k/\delta_x^2$ and $c := k/\delta_y^2$, and the vectors

$$T_j(t) := \begin{pmatrix} T_{0j}(t) \\ T_{1j}(t) \\ \vdots \\ T_{Nj}(t) \end{pmatrix}, \; \partial T_j(t) := \begin{pmatrix} \dfrac{\partial T_{0j}}{\partial t}(t) \\ \dfrac{\partial T_{1j}}{\partial t}(t) \\ \vdots \\ \dfrac{\partial T_{Nj}}{\partial t}(t) \end{pmatrix} \in \mathbb{R}^{(N+1)}, \qquad j = 0, ..., M$$

$$C_j := \frac{2k}{\delta_x} \begin{pmatrix} -c_0(t, y_j) \\ 0 \\ \vdots \\ 0 \\ c_1(t, y_j) \end{pmatrix} \in \mathbb{R}^{(N+1)}, \qquad j = 0, ..., M$$

$$D_0 = -\frac{2k}{\delta_y} \begin{pmatrix} d_0(t, x_0) \\ d_0(t, x_1) \\ \vdots \\ d_0(t, x_N) \end{pmatrix}, \; D_1 = \frac{2k}{\delta_y} \begin{pmatrix} d_1(t, x_0) \\ d_1(t, x_1) \\ \vdots \\ d_1(t, x_N) \end{pmatrix} \in \mathbb{R}^{(N+1)},$$

we can write the whole discretized ODE system as in (2.57):

$$\begin{pmatrix} \partial T_0(t) \\ \partial T_1(t) \\ \vdots \\ \partial T_{M-1}(t) \\ \partial T_M(t) \end{pmatrix} = \begin{pmatrix} A & 2B & 0 & \cdots & 0 \\ B & A & B & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & B & A & B \\ 0 & \cdots & 0 & 2B & A \end{pmatrix} \begin{pmatrix} T_0(t) \\ T_1(t) \\ \vdots \\ T_{M-1}(t) \\ T_M(t) \end{pmatrix} + \begin{pmatrix} C_0 + D_0 \\ C_1 \\ \vdots \\ C_{M-1} \\ C_M + D_1 \end{pmatrix}, \qquad (2.61)$$

which once again exhibits the same sparse, tridiagonal pattern not only in this block structure, but also in the submatrices A, B that compose it. Therefore, it will be particularly advantageous to exploit both this structure and substructure when trying to find numerical solutions for optimal control problems that feature this type of ODE systems.

3 Re-entry modelling

Finding a re-entry trajectory that minimizes heat flux motivates a complex optimal control problem. In this section, we introduce step by step the several aspects of the models that need to be considered in order to compose a re-entry trajectory problem with thermodynamic constraints. This model might not be accurate or realistic enough for detailed spacecraft or operational trajectory design, but it is sufficient for our purpose of applying our methodologies to obtain numerical solutions and demonstrating their applicability and efficiency. Approximations with functions that can be sufficiently differentiated were necessary for some of the described models. We describe as well specific models for the cases of the Sänger hypersonic aircraft concept and the Apollo Command Module obtained from the literature [19, 29, 30, 90] that were used to test our different methodologies with more realistic examples.

3.1 Atmospheric model

An exact atmospheric model that provides values of air density and temperature is needed to calculate the aerodynamic forces and thermodynamic quantities. The US Standard Atmosphere of 1976 has been selected for this purpose [94, 97]. It provides continuous expressions to calculate air temperature for each layer of the atmosphere from 0 to 450 km of altitude, and derivations of pressure, density, and sound speed from temperature. However, in order to preserve the regularity of the functions required to satisfy the necessary optimality conditions that were discussed in Chapter 2, air temperature was approximated with the polynomial

$$T_{air}(h) := \sum_{i=0}^{10} a_i h^i,$$

and an exponential approximation was used to calculate density as

$$\rho(h) = \rho_0 e^{-h/\beta}.$$

Original values from the US 1976 model and our approximations are depicted in Figure 3.1. We obtained a maximum difference between the original and the approximation of 8.15 K for air temperature and 0.04 kg/m^3 for air density.

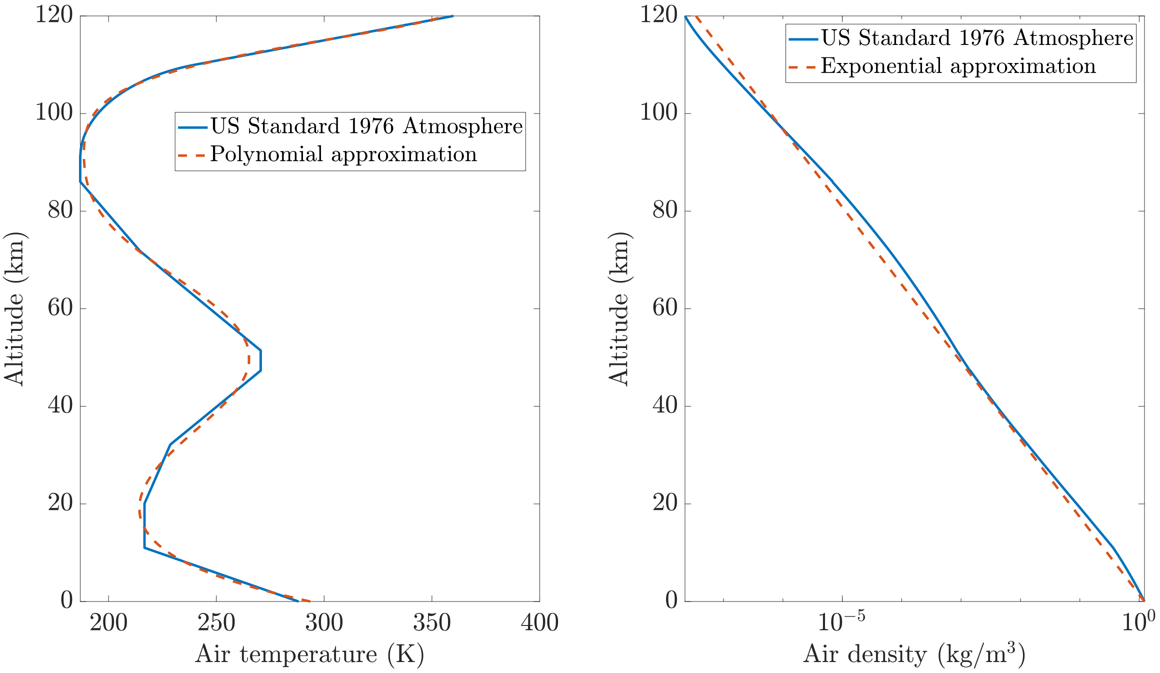

Figure 3.1: Models from US Standard 1976 Atmosphere and approximations

The speed of sound at a certain altitude $c(h)$ can also be calculated from air temperature with the formula

$$c(h) = \sqrt{\gamma R T_{air}(h)},$$

where $\gamma = 1.4$ is the ratio of specific heat of air and $R = 286.9$ $J/(kg \cdot K)$ is the gas constant. The Mach number M can be calculated from altitude and velocity as:

$$M = \frac{v}{c(h)}$$

3.2 Dynamic model

The main external forces acting on an object moving through the atmosphere of the Earth are aerodynamic and gravitational. In order to define the equations of motion for the vehicle, we need to define appropriate models for these forces.

3.2.1 Earth gravitational force

We assume a rotational spheric Earth, given the mean radius and angular velocity [106]:

$$r_E = 6371.01\ km,$$
$$\omega_E = 7.29211 \cdot 10^{-5}\ rad/s.$$

We set gravitational acceleration at sea level to the standard value [97]:

$$g_0 = 9.80665\ m/s^2.$$

According to Newton's law of universal gravitation and considering a one-body problem due to the small mass of the aircraft in relation to the Earth's mass, the gravitational acceleration as a function of altitude can be expressed as

$$g(h) := g_0 \left(\frac{r_E}{r_E + h} \right)^2 .$$

3.2.2 Aerodynamic forces

The aerodynamic forces, lift L and drag D (see Figure 3.2), can be described in terms of the lift and drag coefficients, C_L and C_D respectively, as follows [6, 90, 136]:

$$L = C_L\ q(v,h)\ S_{ref} \tag{3.1}$$
$$D = C_D\ q(v,h)\ S_{ref}, \tag{3.2}$$

where S_{ref} is the reference surface area of the vehicle and $q(v,h)$ is the dynamic pressure

$$q(v,h) = \frac{1}{2}\rho(h)v^2.$$

An exact calculation of the lift and drag coefficients would require deriving them from the angle of attack taking into account several factors, like the shape of the airfoil or the flow conditions. For our purposes, it is sufficient to consider the lift coefficient as a control variable. We resorted to analytical expressions derived from data to express the drag coefficient as a function of the lift coefficient.

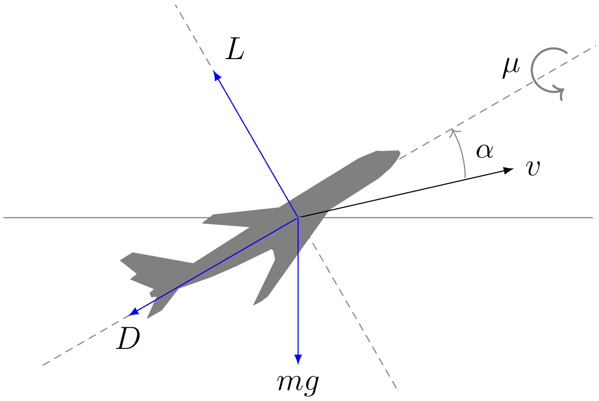

Figure 3.2: Aerodynamic and gravitational forces, and control angles α and μ

We used two different models. The first one is based on the Sänger hypersonic aircraft [19, 90]:

$$C_D(C_L) = C_{D_0} + kC_L^2, \tag{3.3}$$

with $C_{D_0} = 0.017$, $k = 2$. The second one is a linear approximation based on the data representations in [29] for the Apollo command capsule obtained from wind-tunnel data:

$$C_D(C_L) = C_{D_0} - C_L, \tag{3.4}$$

with $C_{D_0} = 1.5$. We can also obtain a similar approximate relation between the angle of attack α and the lift coefficient and therefore, considering C_L as control is equivalent to considering α for the purpose at hand.

3.2.3 Equations of motion

Considering a point mass model and a spherical rotating Earth, we can describe the position of the aircraft in an Earth-centered, Earth-fixed coordinate system with the longitude Θ, the latitude Λ and the altitude h (see Figure 3.3). The movement is described as well by the velocity v, flight-path angle γ and the heading angle χ. We consider a gliding vehicle with no thrust, only influenced by the aerodynamic lift and drag L and D and the gravitational force, and controlled by the lift coefficient C_L and the bank angle μ (see Figure 3.2). The six equations of motion are [91, 136]:

$$\begin{aligned}\dot{v} &= -\frac{D}{m} - g(h)\sin\gamma \\ &\quad + \omega_E^2 (r_E + h)\cos\Lambda(\sin\gamma\cos\Lambda - \cos\gamma\sin\chi\sin\Lambda),\end{aligned} \tag{3.5}$$

$$\begin{aligned}\dot{\gamma} &= \frac{L\cos\mu}{mv} - \left(\frac{g(h)}{v} - \frac{v}{r_E + h}\right)\cos\gamma \\ &\quad + 2\omega\cos\chi\cos\Lambda \\ &\quad + \frac{\omega_E^2 (r_E + h)}{v}\cos\Lambda(\sin\gamma\sin\chi\sin\Lambda + \cos\gamma\cos\Lambda),\end{aligned} \tag{3.6}$$

$$\begin{aligned}\dot{\chi} &= \frac{L\sin\mu}{mv\cos\gamma} - \cos\gamma\cos\chi\tan\Lambda\frac{v}{r_E + h} \\ &\quad + 2\omega_E(\sin\chi\cos\Lambda\tan\gamma - \sin\Lambda) \\ &\quad - \frac{\omega_E^2 (r_E + h)}{v\cos\gamma}\cos\Lambda\sin\Lambda\cos\chi,\end{aligned} \tag{3.7}$$

$$\dot{h} = v\sin\gamma, \tag{3.8}$$

$$\dot{\Lambda} = \frac{v}{r_E + h}\cos\gamma\sin\chi, \tag{3.9}$$

$$\dot{\Theta} = \frac{v}{(r_E + h)\cos\Lambda}\cos\gamma\sin\chi, \tag{3.10}$$

See [133] for details on how these equations are derived.

3.3 Thermodynamic model

Our objective is to reduce heat loads during re-entry trajectories and hence, we require analytical models for aerodynamic heating and temperature evolution that can be coupled with the previously described equations of motion, forming an optimal control problem.

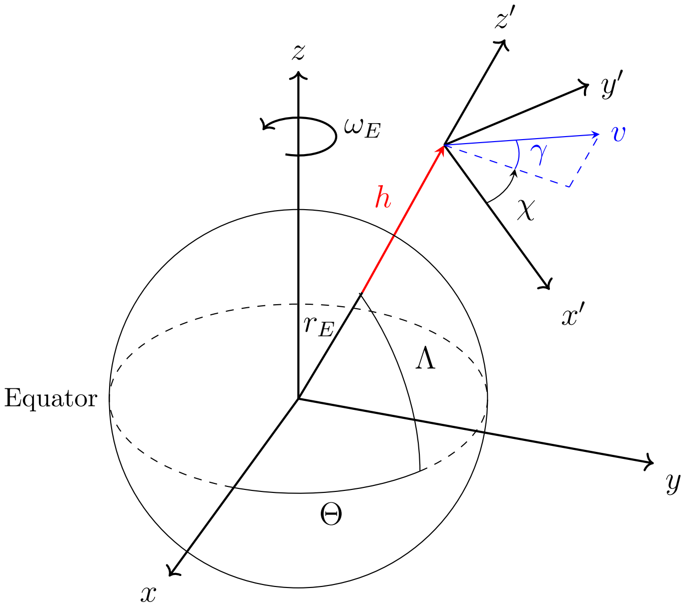

Figure 3.3: Earth coordinate system for system (3.5) - (3.10)

3.3.1 Convective heat flux

For more precise calculations of hypersonic heating over a full body, it is necessary to use CFD or experimental data. CFD high fidelity simulations and experimental tests remain time and resource consuming, and both of these techniques provide heat flux values for a singular scenario after costly calculations. This is insufficient for defining our optimal control problem, since we need heating values to be provided as a continuous function of the aerodynamic variables. Fortunately, several analytical formulas have been developed to estimate heat transfer at the stagnation point [15] and have proven to be sufficiently accurate for similar purposes [21, 30, 104]. The Sutton-Graves correlation expresses convective heat flux at the stagnation point as a function of altitude and velocity [124]:

$$\dot{q}_{conv,sp} = k_E \sqrt{\frac{\rho(h)}{R_N}} v^3. \tag{3.11}$$

Here, R_N is the nose radius and k_E a constant derived for a general gas mixture and planet-specific: In [113] it is calculated as $k_E = 1.7623e^{-4}$. For our general case, we consider the heating only in the stagnation point as it is one of the most critical points during re-entry. In [29], a shcorrection on the stagnation point heat flux to obtain the maximum corner heating $\dot{q}_{conv,max}$ for an Apollo-type capsule using shape parameters is defined with the formula:

$$\frac{\dot{q}_{conv,max}}{\dot{q}_{conv,sp}} = c_1 M + c_2 \alpha + c_3 \frac{R_S}{R_m} + c_4 \theta_N + c_5, \tag{3.12}$$

where R_S, R_M and θ_N are the shape parameters as described in Section 3.5, α is the angle of attack and M is the Mach number. The coefficients $c_1, ..., c_5$ are calculated in [29] using a least squares curve fit for the given data on wind tunnel results, yielding $c_1 = -0.0006$, $c_2 = 0.0185$, $c_3 = -0.5321$, $c_4 = -0.2939$ and $c_5 = 1.3630$.

As for the convective heating in the interior of the vehicle, assuming a constant inner temperature T_{in}, the general convective heat transfer formula can be used:

$$\dot{q}_{conv,in} = \alpha_q (T - T_{in}), \tag{3.13}$$

where α_q is the heat transfer coefficient.

3.3.2 Radiative heat flux

At hypersonic speeds, convective heat flux is not the only thermal load acting on a re-entring vehicle and it is necessary to consider radiative cooling for a more accurate depiction of the heat exchange at the wall. The Stefan-Boltzmann Law describes the power radiated between two bodies (1 and 2) as:

$$\dot{q}_{rad} = \epsilon \sigma (T_1^4 - T_2^4), \tag{3.14}$$

where ϵ is a constant dependent on the material, and $\sigma = 5.67051 \cdot 10^{-8}\ J/(m^2 s K^4)$ is the Stefan-Boltzmann constant. Therefore, on the external boundary, we would have

$$\dot{q}_{rad,air} = \epsilon \sigma (T^4 - T_{air}(h)^4), \tag{3.15}$$

In this model, air temperature in the context of hypersonic flight refers to the temperature after the shock that generates a change in the temperature of the medium. This effect on temperature can be calculated with CFD simulations. For the sake of simplicity, we consider this air temperature as the one obtained from our atmospheric model in Section 3.1. As for the radiative heating in the interior considering constant temperature T_{in}, we have:

$$\dot{q}_{rad,in} = \epsilon\sigma(T^4 - T_{in}^4). \tag{3.16}$$

3.3.3 Temperature evolution

In hypersonic flight, the aerodynamic heating leads to very high temperatures on the surface of the vehicle that can cause severe damages, and as a consequence, it requires a Thermal Protection System (TPS) to get through re-entry safely. In order to study the impact of the heating on the TPS, we consider the one-dimensional heat equation to model the temperature distribution in a perpendicular section to the stagnation point:

$$\rho_{TPS}\; c_{p,TPS} \frac{\partial T(t,x)}{\partial t} = \lambda_{TPS} \Delta T(t,x), \tag{3.17}$$

where $T(t,x)$ is the temperature along the perpendicular section of depth D represented by $x \in [0,D]$, and ρ_{TPS}, $c_{p,TPS}$ and λ_{TPS} are the density, specific heat capacity and heat conductivity of the material of the TPS, respectively. The boundary conditions at the edges of the section are

$$\frac{\partial T}{\partial x}(t,0) = \dot{q}_{conv,sp} - \dot{q}_{rad,air}, \tag{3.18}$$

$$\frac{\partial T}{\partial x}(t,D) = \dot{q}_{conv,in} - \dot{q}_{rad,in}, \tag{3.19}$$

using the previously described formulas (3.11)-(3.16) to calculate the convective and radiative heat flux from the air and the interior of the vehicle. As for the initial condition, we impose a constant temperature

$$T(0,x) = T_{init}, \quad x \in [0,D]. \tag{3.20}$$

There is only a one-way coupling between the dynamic system and the temperature evolution i.e. the temperature does not have any direct influence on the aerodynamic variables or external convective heat flux in our models, but the external heat flux from (3.11) and (3.15) are dependent on the aerodynamic variables and do have a direct influence in the temperature evolution.

In order to test some of our methodologies with a PDE of larger dimensions in Section 5.5, we consider as well a two-dimensional section through the stagnation point into the TPS with an internal cooling system. The domain on the capsule is depicted in Figure 3.4 (the stagnation point is represented on the center of the capsule surface even though its position can vary depending on the angle of attack). We extend (3.17) to a two-dimensional case:

$$\rho_{TPS}\ c_{p,TPS} \frac{\partial T(t,x,y)}{\partial t} = \lambda_{TPS} \Delta T(t,x,y), \tag{3.21}$$

for $(x, y) \in (0, D_x) \times (0, D_y)$, where D_x is the length of the section and D_y the depth. The stagnation point is located at $(0, 0)$, and $(x, y) \in [0, D_x] \times [0, D_y]$ represent the horizontal distance x along the surface, and the depth y through the TPS.

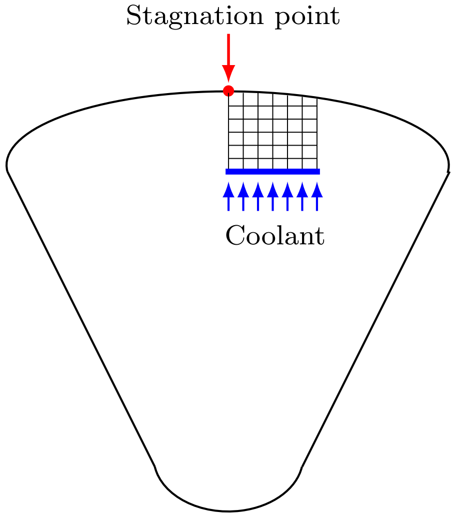

Figure 3.4: 2D section through the TPS from the stagnation point

In this case, boundary conditions for the four edges of the section need to be defined: the external temperature is defined in terms of the convective and radiative external heating as in (3.18). We only have (3.11) to model the convective heating at the stagnation point, due to the complications that using CFD calculations to obtain values for the convective heating along the capsule surface for a whole trajectory would imply. Therefore, given that the heating is maximal at the stagnation point, we consider a linear decrease of the external heating from the stagnation point:

$$\dot{q}_{conv,ext}(t, x) := (1 - rx)\ \dot{q}_{conv,sp}(t), \tag{3.22}$$

where r is a chosen decreasing ratio for the heat flux, and x represents the distance to the stagnation point located at $x = 0$. The idea is to represent the decreasing effect of the heating in the considered area from the stagnation point in order to introduce a variation in the heating for the external boundary, which serves our purpose of modelling a two-dimensional heating effect on a capsule. For the internal boundary, the heat flux $\dot{q}_{cool}(t, x)$ is modelled depending on the temperature of the coolant along the boundary, see the next section for details. Hence, the horizontal boundaries are given by:

$$\frac{\partial T}{\partial y}(x, 0, t) = \dot{q}_{conv,ext}(t, x) - \epsilon\sigma(T(x, 0, t)^4 - T_{air}(h)^4), \tag{3.23}$$

$$\frac{\partial T}{\partial y}(x, y_{max}, t) = \dot{q}_{cool}(t, x). \tag{3.24}$$

For the lateral boundaries, we consider constant derivative conditions:

$$\frac{\partial T}{\partial x}(0, y, t) = 0, \tag{3.25}$$

$$\frac{\partial T}{\partial x}(x_{max}, 0, t) = 0. \tag{3.26}$$

3.4 Active cooling model

For the example based on a two-dimensional heat equation, the temperature decrease given by a cooling system is considered in order to control the temperature in the interior nodes. From [3, 88], we consider an active cooling system that uses cryogenic fuels to compensate the effects of the aerodynamic heating and to support the TPS. In [3, 4] several coolants are compared, showing liquid hydrogen to be a clearly superior choice in terms of cooling and saving mass, and being able to reach temperatures as low as 14 K.

We considered as control the temperature of the cooling system $T_{cool}(t, x)$ depending on the position along the section, and we use equations (3.13) and (3.14) to model the heat flux interaction between the two-dimensional section of the TPS and the coolant:

$$\dot{q}_{cool}(t, x) := \alpha_q(T(t, x, D_y) - T_{cool}(t, x)) + \epsilon\sigma(T(t, x, D_y)^4 - T_{cool}(t, x)^4), \tag{3.27}$$

for $x \in [0, D_x]$. The goal is to minimize the use of coolant while keeping the temperature at the interior nodes of the section under a certain threshold. From [3], coolant mass rate is modelled as:

$$\dot{m} = A_c \, \dot{q}_{cool},$$

where A_c is a constant depending on the area of the cooling section and the characteristics of the coolant. Therefore, since the mass rate of the coolant is directly proportional to the heat flux, minimizing both quantities is equivalent. The heat flux $\dot{q}_{cool}$ is mainly influenced by the controllable cooling temperature T_{cool} in (3.27); specifically, it decreases when T_{cool} remains unchanged from its nominal value, and it increases when T_{cool} decreases, which means there is an activation of the cooling system in order to lower T_{cool}. In other words, minimizing the use of coolant would be equivalent to maximizing the temperature of the cooling system along the trajectory.

3.5 Shape model

A realistic, detailed shape optimization is out of the scope of this work. However, given the simplicity of the Apollo capsule shape that can be expressed in terms of only a few parameters, we included these in our model focused on the Apollo command capsule to analyze the effect of including a few shape parameters in our trajectory optimization.

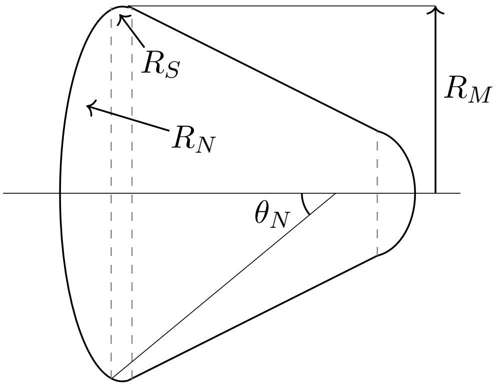

Figure 3.5: Parametrized capsule shape, compare [29]

The Apollo capsule shape approximate parametrization developed in [29] divides the capsule into four analytical shapes defined by 5 parameters (see Figure 3.5). We only considered the three parameters that define the analytical frontal shape of the Apollo capsule: nose radius R_N, side radius R_S and mid radius R_M. These are all used to calculate the maximum heat flux with the formula (3.12). The nose sphere angle θ_N is also present in (3.12), but it can be derived from these three parameters as:

$$\theta_N = \arcsin\left(\frac{R_M - R_S}{R_N - R_S}\right).$$

The reference surface area S_{ref} can be calculated from the nose radius and the nose sphere angle with a standard formula to calculate the area of a spherical cap:

$$S_{ref} = 2\pi R_N^2(1 - \cos\theta_N). \tag{3.28}$$

This means the shape parameters can also have an influence on the trajectory, since S_{ref} is present in the calculation of lift and drag, see (3.1) and (3.2).

4 Optimal re-entry trajectories with OCPID-DAE1

This chapter is devoted to finding optimal re-entry trajectories with minimum heating numerically, using a reduced discretization approach. We discuss three different problems: a simple re-entry trajectory problem with the heat load as objective function, considered in two different scenarios for different vehicles; the same problem with the addition of a parametrization of an Apollo-type capsule in order to optimize shape parameters as well; and finally, a coupled ODE-PDE featuring a heat equation that models the temperature evolution, with different objective functions based on the temperature and the heat flux being considered for comparison. These results have been expanded from those presented in [99].

OCPID-DAE1 [40] is a Fortran package designed to solve optimal control problems and parameter identification problems subject to ordinary differential equations and differential algebraic equations, control and state constraints, and boundary conditions. It implements a direct multiple shooting method with several ODE integrator options (e.g. Euler, Heun, classic Runge-Kutta methods...) and control approximation by B-splines in order to apply a reduced discretization approach. It can also perform adjoint estimation based on the conditions of the local minimum principle. OCPID-DAE1 uses the optimization software `sqpfiltertoolbox`, an implementation of an SQP method for general constrained NLPs. It offers a choice of line-search strategies with merit functions or filter techniques in order to achieve convergence from arbitrary starting points. Derivatives can be provided by the user or are approximated automatically by finite difference approximations, and Hessians are calculated with BFGS updates in every iteration. The code is designed for small-scale to medium-scale problems with dense Jacobian and Hessian matrices.

OCPID-DAE1 is able to solve optimal control problems where a small number of controls is involved robustly and efficiently. With the following results, it proves to be a useful tool for the purpose of finding optimal re-entry trajectories with minimum heating, and testing different models and problems. It also proves to be able to deal with coupled ODE-PDE problems involving the heat equation, which can be challenging due to the large, complex structure they present. These results will also be used in the next chapter to define optimal external heating for larger cases of the heat equation involving a larger number of controls.

4.1 Re-entry trajectory problem with minimum heating

Referring to the models described in Sections 3.2 and 3.3, we formulate a simple re-entry trajectory optimal control problem with the objective of minimizing the convective heat flux $\dot{q}_{conv}$ along the trajectory as follows:

Problem 4.1 (Re-entry trajectory OCP). *Minimize*

$$\int_0^{t_f} \dot{q}_{conv}(t)\ dt = q_{conv}(t_f) \tag{4.1}$$

w.r.t. t_f, $C_L(t)$ and $\mu(t)$, subject to the differential equations (3.5)-(3.10) and (3.11) and the constraints

$$(v(t_0), \gamma(t_0), \chi(t_0), h(t_0), \Lambda(t_0), \Theta(t_0), q_{conv}(t_0)) = (v_0, \gamma_0, \chi_0, h_0, \Lambda_0, \Theta_0, 0), \tag{4.2}$$

$$h(t_f) \leq h_{max} \tag{4.3}$$

$$q(v(t), h(t)) \leq Q_{max}, \qquad \forall t \in [t_0, t_f], \tag{4.4}$$

$$C_{L,min} \leq C_L(t) \leq C_{L,max}, \qquad \forall t \in [t_0, t_f], \tag{4.5}$$

$$\mu_{min} \leq \mu(t) \leq \mu_{max}, \qquad \forall t \in [t_0, t_f], \tag{4.6}$$

$$t_{f,min} \leq t_f \leq t_{f,max}. \tag{4.7}$$

Here, the states are the six position variables $(v(t), \gamma(t), \chi(t), h(t), \Lambda(t), \Theta(t))$ and the heat flux $\dot{q}_{conv}(t)$, which we strategically placed among the differential equations in order to obtain the value of its integral along the trajectory for the objective function, also known as the heat load q_{conv}. The initial conditions (4.2) are given by the initial orbital position of the vehicle and considering $q_{conv}(t_0) = 0$, and (4.3) ensures the re-entry condition, i.e. that the final altitude is close to the surface of the Earth. The constraint (4.4) on the dynamic pressure $q(v, h)$ is included as well to ensure that the vehicle does not surpass its maximum allowed mechanical stress, commonly known as max q.

As a first step to test our models and the application of OCPID-DAE1, we solved Problem (4.1) in two different initial scenarios: a baseline scenario based on the hypersonic Sänger aircraft concept obtained from [19, 41, 90], from a lower altitude of 33.9 km (Case 1), and a scenario based on the Apollo capsule obtained from [29, 60, 61] from a typical Low-Earth-Orbit (LEO) at 120 km of altitude (Case 2). Table 4.1 compiles the values of the initial values, vehicle-specific constants required in the models from Chapter 3, and bounds on controls and parameters. For calculating the drag coefficient from the lift coefficient, model (3.3) was used for Case 1, and (3.4) for Case 2, according to the vehicle considered for each case. The initial latitudes and longitudes correspond to a position near Bayreuth, Germany, for Case 1, and over the Pacific Ocean for Case 2. The mass m, nose radius R_N and reference area S_{ref} were also chosen appropriately for each vehicle. We chose a very restrictive max q value for Case 2, in order to make sure that the state constraint became active and to see its effect on the solution; as orientative values, reports [60, 61] show the evolution of the dynamic pressure during re-entry for different Apollo missions reaching values over 8 kN/m^2 and 24 kN/m^2, respectively.

Case 1				Case 2			
v_0	2.15 km/s	h_0	33.9 km	v_0	7.83 km/s	h_0	120.0 km
γ_0	2°	Λ_0	49.57°	γ_0	-2°	Λ_0	-23.75°
χ_0	130°	Θ_0	11.35°	χ_0	40°	Θ_0	225.5°
$C_{L,min}$	0.01	$C_{L,max}$	0.18326	$C_{L,min}$	0	$C_{L,max}$	0.5
μ_{min}	-45°	μ_{max}	45°	μ_{min}	-45°	μ_{max}	45°
$t_{f,min}$	100 s	$t_{f,max}$	500 s	$t_{f,min}$	500 s	$t_{f,max}$	2000 s
h_f	0.5 km	Q_{max}	60 kN/m^2	h_f	2.0 km	Q_{max}	7 kN/m^2
m	115000 kg	S_{ref}	305 m^2	m	5897 kg	S_{ref}	39.44 m^2
R_N	0.5 m			R_N	4.69 m		

Table 4.1: Constants and parameters for Problem 5.3

For these problems and the following ones, the number of time nodes for the discretization was set to $N = 200$, the order of the B-splines (2.31) used to discretize the control was set to $k = 1$ (piecewise constant approximation), and the classic Runge-Kutta method was chosen as ODE integration method. In the SQP method, a line-search with the augmented Lagrangian as the merit function was selected as globalization strategy, and the feasibility and optimality tolerance were set to 10^{-8} and 10^{-6}, respectively. For Case 1, the software converged to a solution in 155 SQP iterations and took 9.04 seconds of CPU time in total; for Case 2, it converged in 99 iterations and took 7.44 seconds. It should be pointed out that the number of iterations and, correspondingly, the CPU times are highly dependent on how close the initial guess is to the solution.

Figures 4.1 and 4.3 depict the obtained optimal trajectories in terms of latitude, longitude and altitude (h, Λ and Θ) for Cases 1 and 2, respectively. The obtained minimum objective function values were $\int \dot{q}_{conv} = 3778.13\ J/cm^2$ for Case 1, and $\int \dot{q}_{conv} = 5112.89\ J/cm^2$ for Case 2. For reference on Case 2, the Apollo TPS experience report [102] indicates heat loads over 20000 J/cm^2. The minimum heat flux trajectory is depicted in Figure 4.2 for Case 1 and in Figure 4.4 for Case 2, along with velocity v (the main component that influences the heat flux along with air density) and controls C_L and μ. The optimal final times are $t_f = 506.57\ s$ for Case 1 and $t_f = 792.68\ s$ for Case 2.

The adjoints λ_i associated to the states $i = v, \gamma, \chi, h, \Lambda$ are plotted for each of the cases in Figures 4.5 and 4.6. The two remaining states, Θ and $\dot{q}_{conv}$, yield constantly 0 adjoints due to the fact that they do not appear in the dynamic functions in (3.5)-(3.10). The state constraint on the dynamic pressure does not become active in the solution for Case 1, which means its associated multiplier is constantly 0 due to the complementarity condition. In Case 2 it does become active around $t = 428\ s$, which allows for the multiplier to change its value at this point, as depicted in Figure 4.7. This means as well that there can be jumps in the adjoints, such as the one that can be appreciated in λ_h at the time the state constraint becomes active.

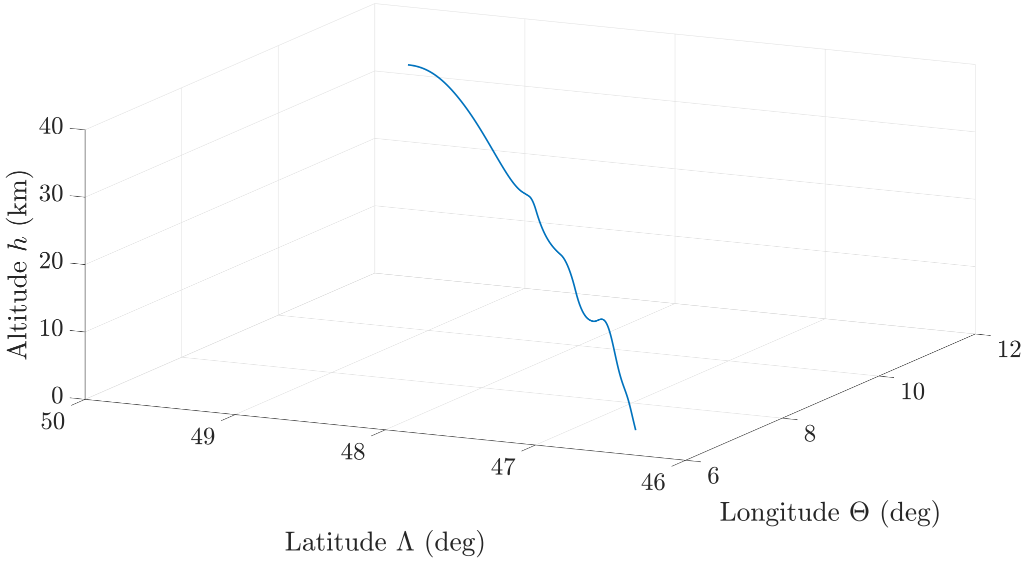

Figure 4.1: Optimal re-entry trajectory for Case 1

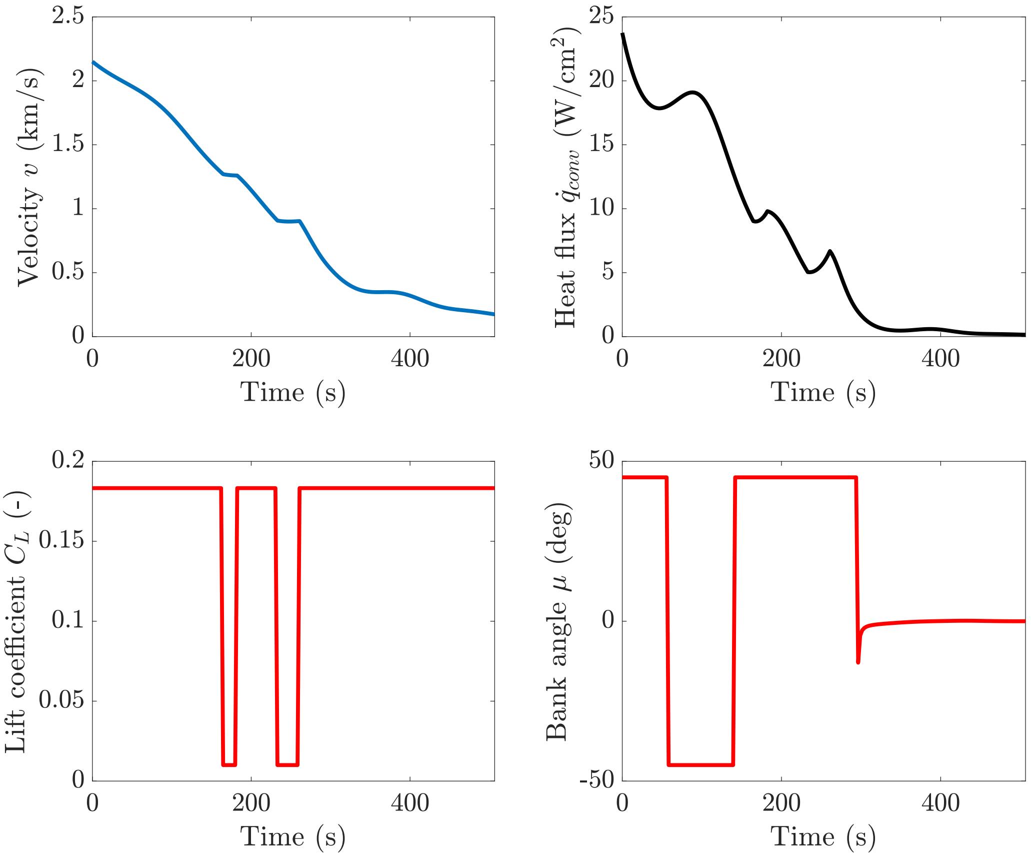

Figure 4.2: Velocity, heat flux and optimal controls for Case 1

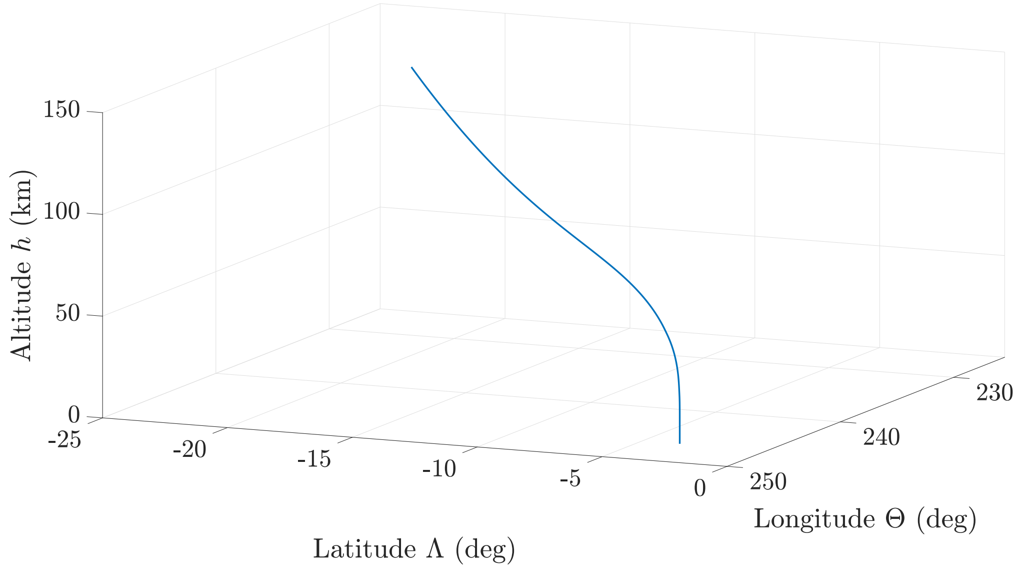

Figure 4.3: Optimal re-entry trajectory for Case 2

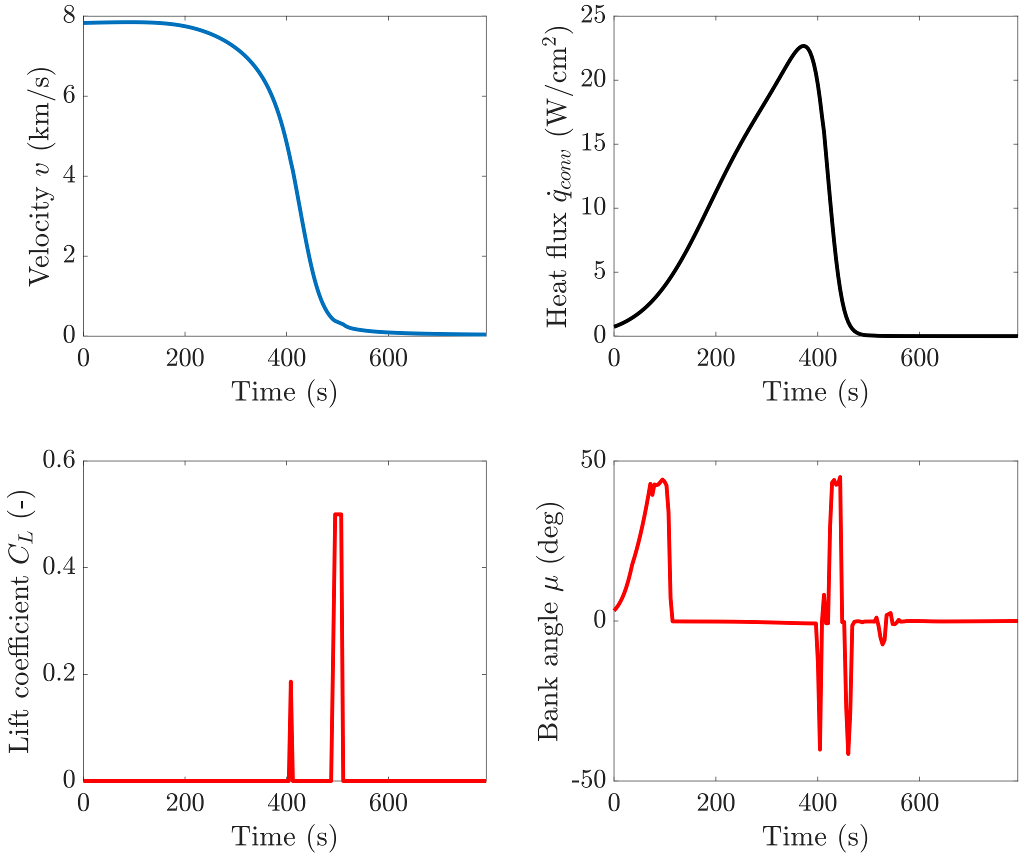

Figure 4.4: Velocity, heat flux and optimal controls for Case 2

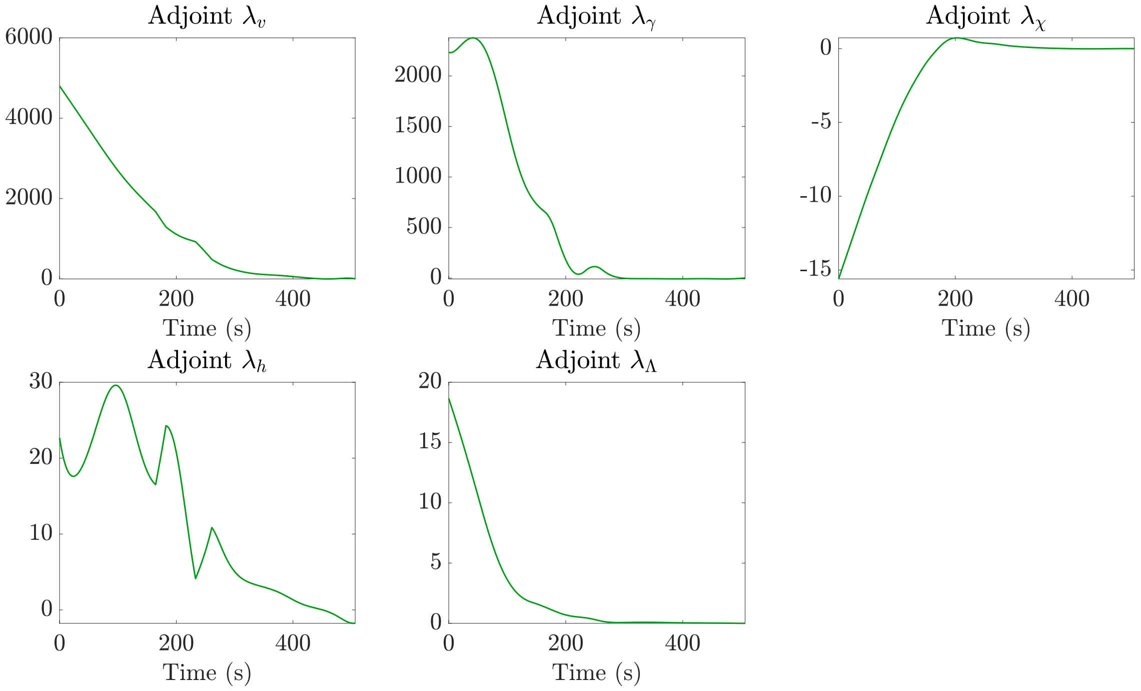

Figure 4.5: Adjoints for states v, γ, χ, h and Λ for Case 1

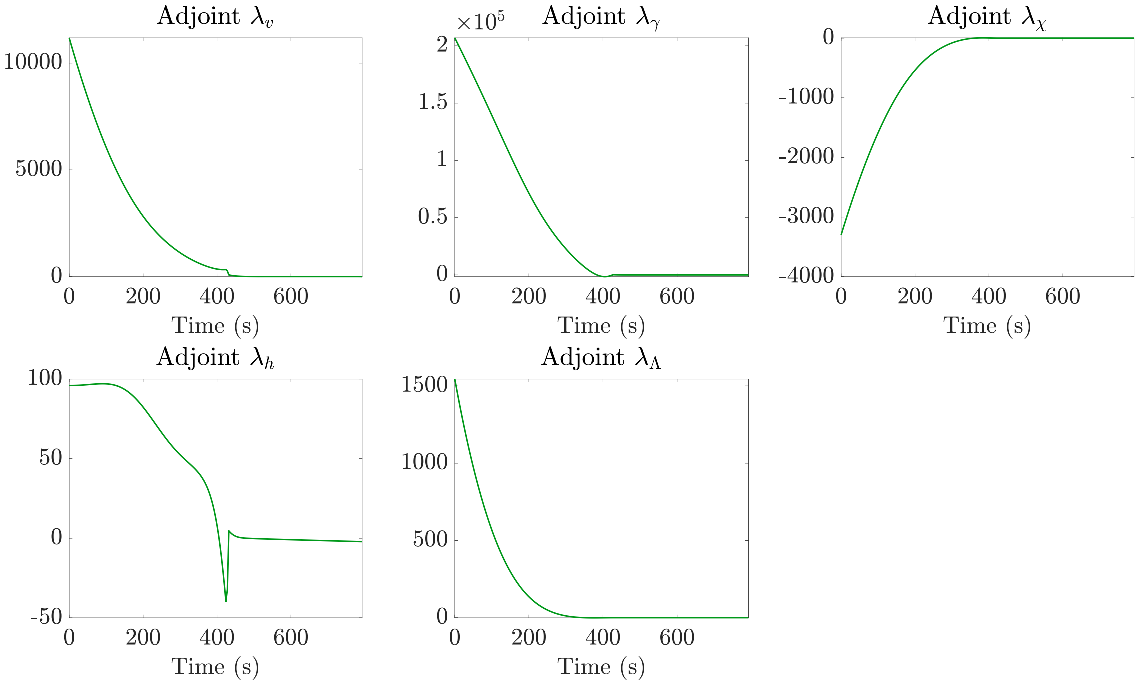

Figure 4.6: Adjoints for states v, γ, χ, h and Λ for Case 2

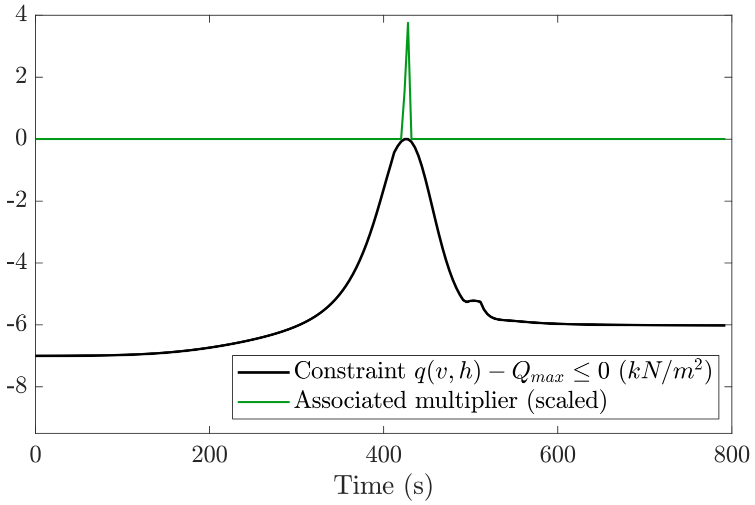

Figure 4.7: Dynamic pressure state constraint and associated multiplier for Case 2

4.2 Parametric shape optimization

The influence of the vehicle shape on the heat flux is already evident in the previous test problems, since the nose radius is present in the Sutton-Graves formula used to calculate the heat flux at the stagnation point. We tested an extension of the previous Case 2 where the capsule shape is described by three parameters, as exposed in Section 3.5: the nose radius R_N, mid radius R_M and shoulder radius R_S. The convective heat flux $\dot{q}_{conv}$ is now calculated with the formula (3.12) that involves all three of these parameters (in what follows, we refer to it as $\dot{q}_{conv,max}$ to differentiate it from the convective heat flux at the stagnation point $\dot{q}_{conv,sp}$ used in the previous section), and the surface area S_{ref} needed to obtain lift and drag for the dynamic equations is now calculated via (3.28).

We solved Problem 4.1 with these modifications and imposing bounds on the three parameters R_N, R_M and R_S. Additionally, a constraint on the weighted sum of the three shape parameters was considered:

$$w_1(R_N - b_1) + w_2(R_M - b_2) + w_3(R_S - b_3) \leq P_{max} \tag{4.8}$$

where the weights w_1, w_2, w_3 and biases b_1, b_2, b_3 are chosen to normalize the three parameters: $w_1 = 0.25$, $b_1 = 3.0$, $w_2 = 1.0$, $b_2 = 2.0$, $w_3 = 2.63$ and $b_3 = 0.02$; and

$P_{max} \leq 3$ is a chosen constant. The constraint can be imposed in order to evaluate how the parameters would have to be decreased should there be a constraint on the area or volume of the capsule, for example; in other words, to find out what parameters are more important for the minimization of the heat flux and which ones can be decreased if needed. The bounds and optimal parameters without the constraint and with the constraint with $P_{max} = 1.5$ are shown in Table 4.2.

	Min value	Max value	Optimal value	Optimal value with constraint (4.8)
R_N	3.0 m	7.0 m	7.0 m	5.0 m
R_M	2.0 m	3.0 m	3.0 m	3.0 m
R_S	0.02 m	0.4 m	0.02 m	0.02 m

Table 4.2: Bounds and optimal shape parameters for Problem 5.3

From these results, we can draw several interesting conclusions. The optimal parameters in the case without the constraint show that, as could be expected, when minimizing heat flux a larger capsule with a small shoulder is preferable. Hence, both the nose radius R_N and mid radius R_M that define the size of the frontal surface of the capsule are at their upper bounds, and the shoulder radius at its lower bound. When including the constraint, we see that both R_M and R_S remain unchanged, and only R_N is decreased: this means that the contribution to the heat flux is higher when R_M and R_S vary from their optimal value, and lower in the case of R_N.

The inclusion of the shape parameters in the formulation leads to a different optimal re-entry trajectory as well. In Figure 4.8, the optimal controls for the shape optimization problem are shown along with those obtained for the original problem (Case 2 from the previous section) with a very noticeable variation. Consequently, the states also vary from the solution for the original problem. Both solutions and their respective adjoints are pictured in Figures 4.9 and 4.10. Note that we compare the heat flux at the stagnation point $\dot{q}_{conv,sp}$ for both solutions, but the objective function for the shape optimization problem was $\int \dot{q}_{conv,max}$, where $\dot{q}_{conv,max}$ is calculated from $\dot{q}_{conv,sp}$ and the shape parameters as in (3.12); it is plotted separately as well for comparison.

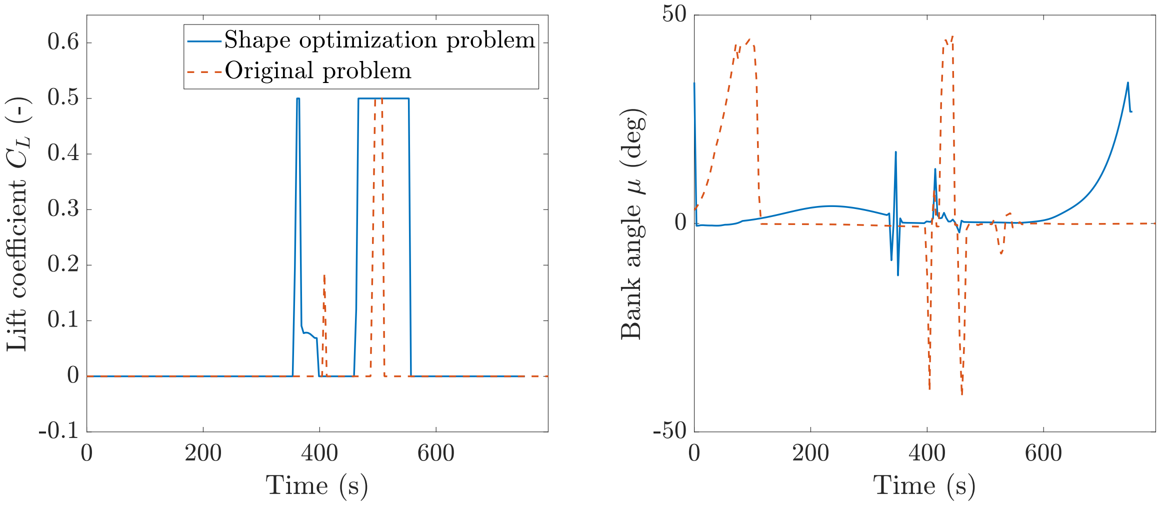

Figure 4.8: Optimal controls for the shape optimization problem and for Case 2 from Section 4.1

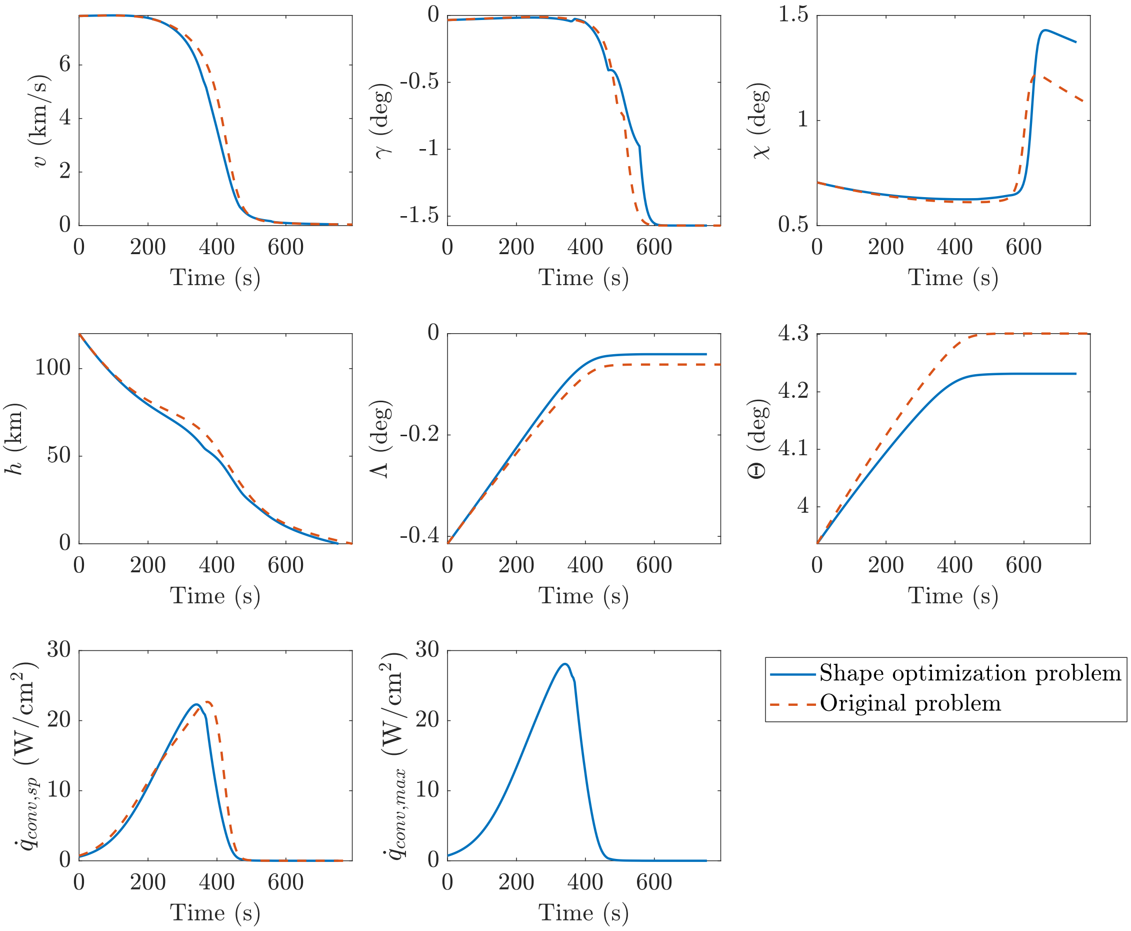

Figure 4.9: Solution for the shape optimization problem and for Case 2 from Section 4.1

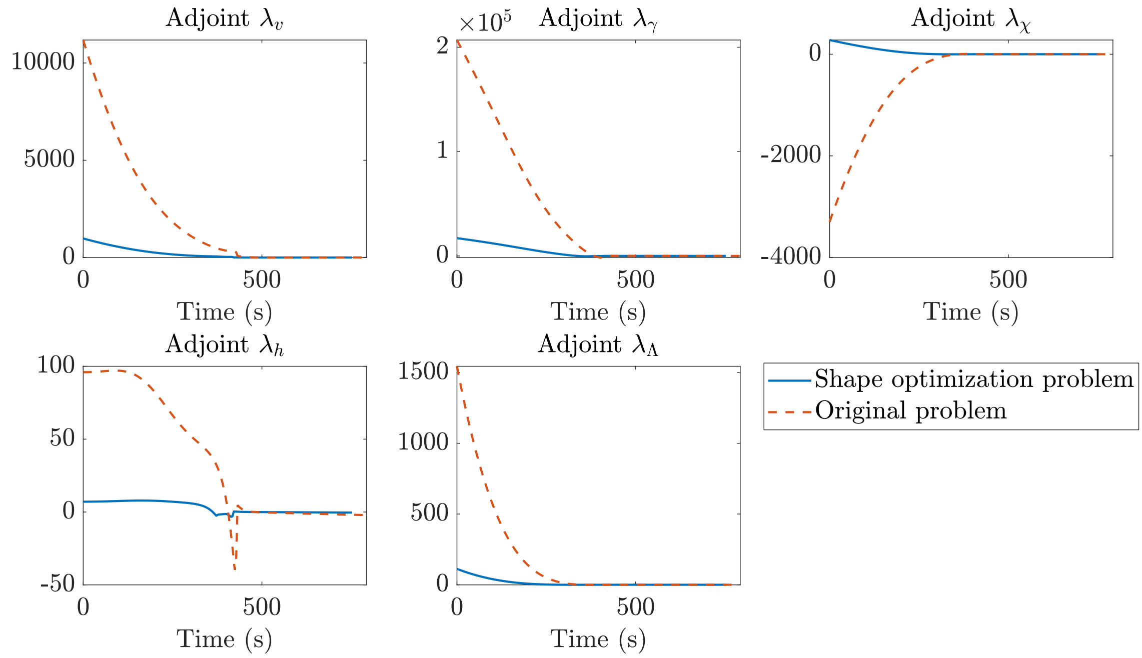

Figure 4.10: Adjoints for the shape optimization problem and for Case 2 from Section 4.1

OCPID-DAE1 found a solution for the shape optimization problem in 451 iterations and 28.49 s of CPU time: we reiterate the fact that these are highly dependent on a good initial guess, specially for parameters. The minimum objective function value resulted in $\int \dot{q}_{conv,max} = 5710.16\ J/cm^2$, larger than the obtained $\int \dot{q}_{conv,sp} = 5112.89\ J/cm^2$ for the original problem as expected. The obtained final time, $t_f = 752.49\ s$, also differed from the obtained $t_f = 792.68\ s$ for the original problem.

4.3 ODE-PDE coupled optimal control problem

After obtaining an optimal trajectory with minimum heat flux, the next desirable goal would be to involve the temperature evolution on the aircraft in the optimization process. OCPID-DAE1 is able to solve a range of different problems with different challenging aspects and yield fast and robust numerical solutions, as shown with the previous test problems. Its main limitation is that a larger number of controls would lead to a very dense, larger optimization problem that can severely affect the performance of an SQP method for small, dense problems. For this reason, PDE problems of a larger dimension and with a larger number of controls may pose a problem for this methodology and therefore, they are reserved for the full discretization approach in the next chapter. Therefore, we limit ourselves to consider an ODE-PDE coupled optimal control problem with the re-entry trajectory dynamics and a one-dimensional heat equation to model the temperature evolution.

We extend again Case 2 from Section 4.1 with the addition of temperature to the states, coupling the PDE

$$\rho_{TPS}\, c_{p,TPS} \frac{\partial T(t,x)}{\partial t} = \lambda_{TPS} \Delta T(t,x), \quad t \in (0,t_f),\ x \in (0,D) \tag{4.9}$$

to the ODE dynamic system, along with the constraints

$$\frac{\partial T}{\partial x}(t,0) = \dot{q}_{conv,sp} - \dot{q}_{rad,air}, \tag{4.10}$$

$$\frac{\partial T}{\partial x}(t,D) = \dot{q}_{conv,in} - \dot{q}_{rad,in}, \tag{4.11}$$

$$T(0,x) = T_{init} \tag{4.12}$$

where all the heat fluxes are defined as in formulas (3.11)-(3.16). We describe briefly the coupling betweeen the ODE and PDE system. Given the controls $(C_L(t), \mu(t))$ that we intend to optimize, we can obtain the position and direction of the vehicle by solving the ODE system (3.5)-(3.10), which allows us to obtain the heat flux as well. Given these ODE variables, we can define the external boundary for the PDE and solve it to obtain $T(x,t)$. We have a one-way coupling so far, which means the temperature is not really involved in the optimization process and we only have a simulation of its evolution given by the solution of the PDE. However, if the temperature is involved in the objective function or constraints as well, this will have an influence on the optimal controls which defines a fully coupled system, see Figure 4.11.

Following the method of lines along the grid $x_i = i\delta,\ i = 0, ..., M$ with $M \in \mathbb{N}$ and $\delta = D/M$ to discretize (4.9), we obtain the collection of states $(T_0, ..., T_M)$ that represent the temperature at each spatial node x_i, being $x_0 = 0$ the stagnation point and $x_M = D$ at the most inner layer of the TPS. With the discretization of (4.9) and the constraints (4.10)-(4.12) and defining the heat diffusivity $k := \lambda_{TPS}/(\rho_{TPS}\, c_{p,TPS})$ and $T(t) := (T_0(t), ..., T_M(t))$, the problem in question would read as:

Problem 4.2 (Discretized coupled re-entry trajectory problem)**.** *Minimize*

$$J(t_f, \dot{q}_{conv,sp}(t), T(t)) \tag{4.13}$$

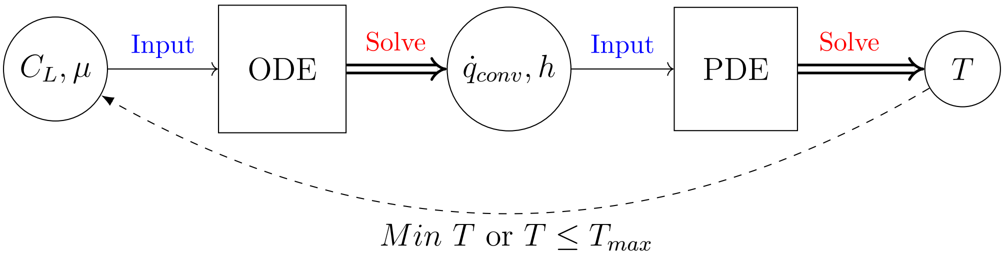

Figure 4.11: Coupling dependencies for a re-entry problem with heat equation

w.r.t. t_f*,* $C_L(t)$ *and* $\mu(t)$*, subject to the differential equations (3.5)-(3.10), (3.11) and*

$$\dot{T}_i = \frac{k}{\delta^2}\left(T_{i-1}(t) - 2T_i(t) + T_{i+1}(t)\right), \quad i = 1, ..., M-1, \tag{4.14}$$

$$\dot{T}_0 = \frac{k}{\delta^2}\left(\frac{\delta}{\lambda_{TPS}}\left(\dot{q}_{conv,sp}(t) - \epsilon\sigma(T_0(t)^4 - T_{air}(h(t))^4)\right) - T_0(t) + T_1(t)\right), \tag{4.15}$$

$$\dot{T_M} = \frac{k}{\delta^2}\left(T_{M-1}(t) - T_M(t) - \frac{\delta}{\lambda_{TPS}}(\alpha_q(T_M(t) - T_{in}), -\epsilon\sigma(T_M(t)^4 - T_{in}^4).)\right), \tag{4.16}$$

and the constraints (4.2)-(4.7) and

$$T_i(0) = T_{init}, \quad i = 0, ..., M. \tag{4.17}$$

Here, $J(t_f, \dot{q}_{conv,sp}(t), T(t))$ is used to denote the different objective functions we tested for this problem. In the previous problems in this chapter, our models have been limited to minimizing the external convective heat flux. The inclusion of the temperature in this section allows for other objective functions to be considered. In particular, we considered the following cases:

- Minimizing the heat load, as in previous problems (Solution 1):

$$J(t_f, \dot{q}_{conv,sp}(t), T(t)) = \int_{t_0}^{t_f} \dot{q}_{conv,sp}(t)\ dt \tag{4.18}$$

- Minimizing the maximum convective heat flux (Solution 2):

$$J(t_f, \dot{q}_{conv,sp}(t), T(t)) = \max_{t\in[t_0,t_f]} \dot{q}_{conv,sp}(t) \tag{4.19}$$

- Minimizing the maximum external temperature (Solution 3):

$$J(t_f, \dot{q}_{conv,sp}(t), T(t)) = \max_{t\in[t_0,t_f]} T_0(t) \tag{4.20}$$

- Minimizing the maximum internal temperature (Solution 4):

$$J(t_f, \dot{q}_{conv,sp}(t), T(t)) = \max_{t \in [t_0, t_f]} T_M(t) \tag{4.21}$$

Defining the maximum of one of the states $x(t)$ along the trajectory as the objective function is accomplished by defining an auxiliary parameter p, considering p as the objective function to minimize, and adding to the problem the constraint

$$x(t) \leq p, \quad t \in [t_0, t_f]. \tag{4.22}$$

Table 4.3 compiles all the constants and initial values needed to define the dynamic system with the discretized PDE to complete those from Table 4.1. All constants related to heat transfer are obtained from [80, 136] with the intention of reproducing the case of the Apollo capsule and using realistic values: specifically, the values that describe the physical properties of the TPS are obtained from [80], a NASA report that performs a re-entry thermal analysis for the Apollo capsule. We also relaxed the value of Q_{max} to a more realistic 8 kN/m^2 as suggested by the dynamic pressure values in [60, 61].

Constant	Value	Constant	Value
ρ_{TPS}	528.6 kg/m^3	ϵ	0.8
$c_{p,TPS}$	2742.35 J/kg K	σ	5.67 $\cdot 10^{-8}$ J/m^2 s K^4
$\lambda_{p,TPS}$	0.242 W/K m	α_q	35 J/m^2 s K
T_{in}	300 K	T_{init}	300 K
D	0.07 m		

Table 4.3: Constants and parameters for Problem 4.2

We considered $M = 10$ spatial nodes and solved the problem for the aforementioned four different objective functions to calculate four solutions for Problem 4.2. The minimized objective values and performance results can be found in Table 4.4, as well as the optimal values for the t_f parameter. Solutions 2 and 3 took a considerably larger CPU time, this is probably due to the initial guess being further from them and closer to Solutions 1 and 4. Solution 2 required the least number of iterations

	Obj. function	Value	Optimal t_f	Iterations	CPU time
Solution 1	$\int \dot{q}_{conv,sp}$	5101.04 J/cm^2	740.54 s	65	12.90 s
Solution 2	max $\dot{q}_{conv,sp}$	13.50 W/cm^2	1117.69 s	51	58.07 s
Solution 3	max T_0	1252.38 K	1090.43 s	101	96.37 s
Solution 4	max T_M	331.68 K	740.39 s	52	13.80 s

Table 4.4: Results for Problem 4.2

despite its larger CPU time: this is due to some of these iterations in the SQP solver requiring a larger number of iterations to solve the QP.

The optimal controls for each trajectory are represented in Figure 4.12, where the difference between the four solutions can be most appreciated. However, as shown in Figures 4.13 and 4.14, the obtained trajectories for Solutions 1 and 4 barely differ from one another, and they are also quite similar for Solutions 2 and 3. Therefore, minimizing the heat load and minimizing the internal temperature seem to be related, as well as minimizing the maximum heat flux and the maximum external temperature. This could be expected from the fact that the external temperature is mostly influenced by the heat flux in (4.15). However, the fact that the optimization of the maximum temperature at the external node and the internal node yields a different optimal re-entry trajectory implies that the involvement of the heat equation is relevant in this problem, and that it does not suffice to minimize the external heating if our goal is to minimize the temperature increase at the inner structure of the capsule due to safety reasons.

The full temperature evolution along the one-dimensional section is depicted in Figure 4.15. The external temperature is the highest for most of the time interval until there is a sudden decrease in the heat flux (see Figure 4.14), after which the accumulated heat load in the inner nodes becomes higher than in the external one.

Finally, all the adjoints to the states that were non-zero are plotted in Figure 4.16. It is interesting to see how the adjoints to the temperature states $T_0, ..., T_{10}$ are constantly zero for Solutions 1 and 2 since they were not involved in the objective function and as previously mentioned, only a simulation of the temperature evolution was performed along with the heat flux optimization. However, in the case of Solutions 3 and 4, the temperatures at all the nodes have non-zero adjoints which means the PDE is fully involved in the optimization. The state constraint on the dynamic pressure does not become active for any of the solutions.

In conclusion, these solutions prove that OCPID-DAE1 can efficiently deal with re-entry trajectory optimal control problems and find solutions to problems with different initial conditions, nonlinear constraints, objective functions and parameters to optimize. In particular, it can efficiently find solutions to coupled ODE-PDE problems where the PDE is fully involved in the optimization. The dimensions of the PDE could be increased as long as the number of controls does not increase. However, a good initial guess for the controls and parameters can be crucial to find these solutions efficiently, and a bad guess can lead to a larger number of iterations and a considerable increase in the CPU times.

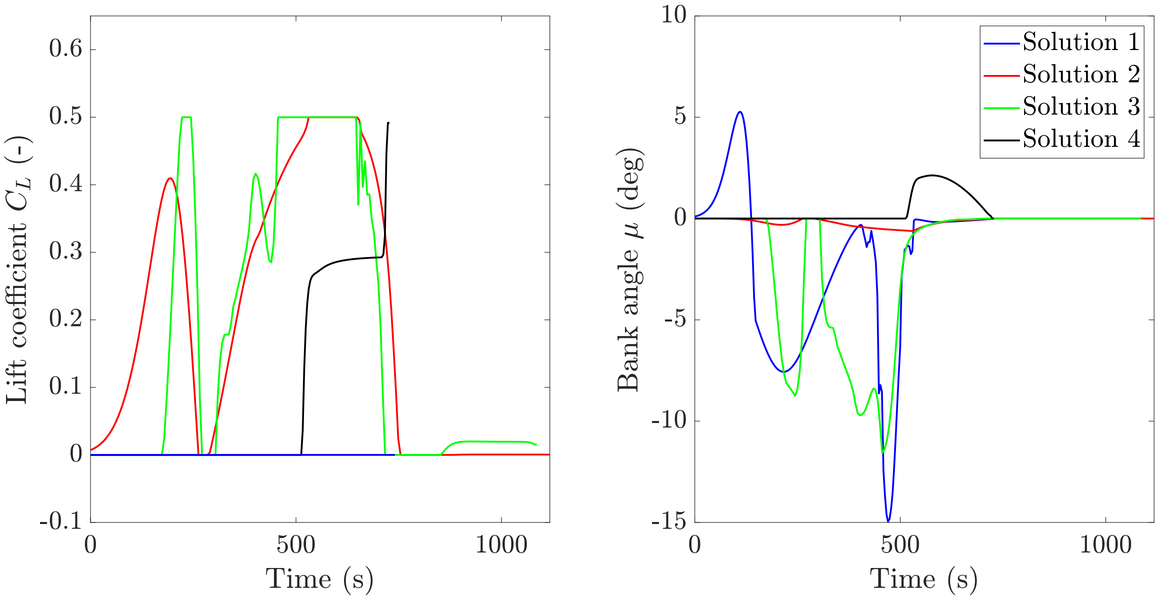

Figure 4.12: Optimal controls for Solutions 1-4 to Problem 4.2

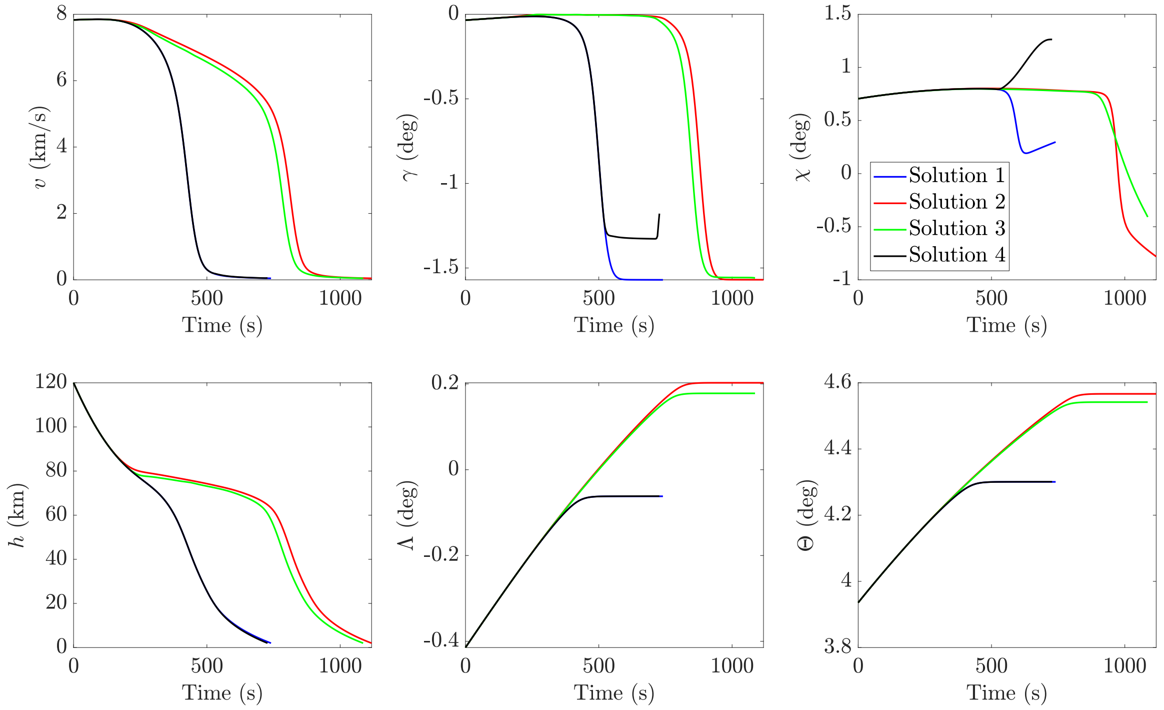

Figure 4.13: Trajectory for Solutions 1-4 to Problem 4.2

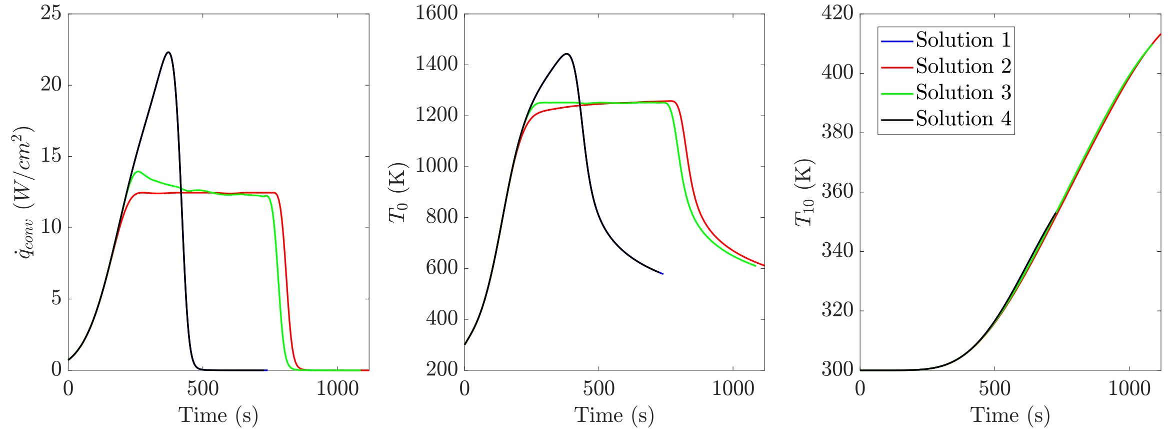

Figure 4.14: Heat flux and boundary temperatures for Solutions 1-4 to Problem 4.2

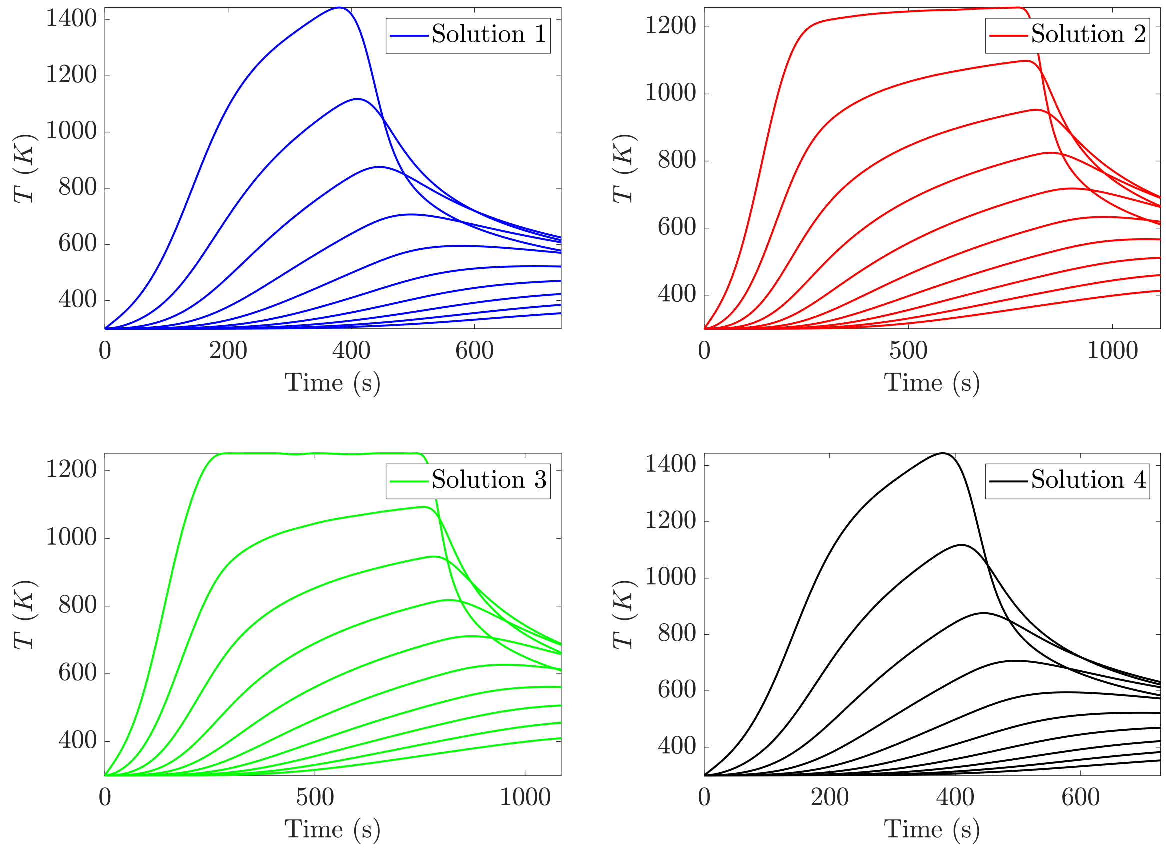

Figure 4.15: Temperature evolution for Solutions 1-4 to Problem 4.2

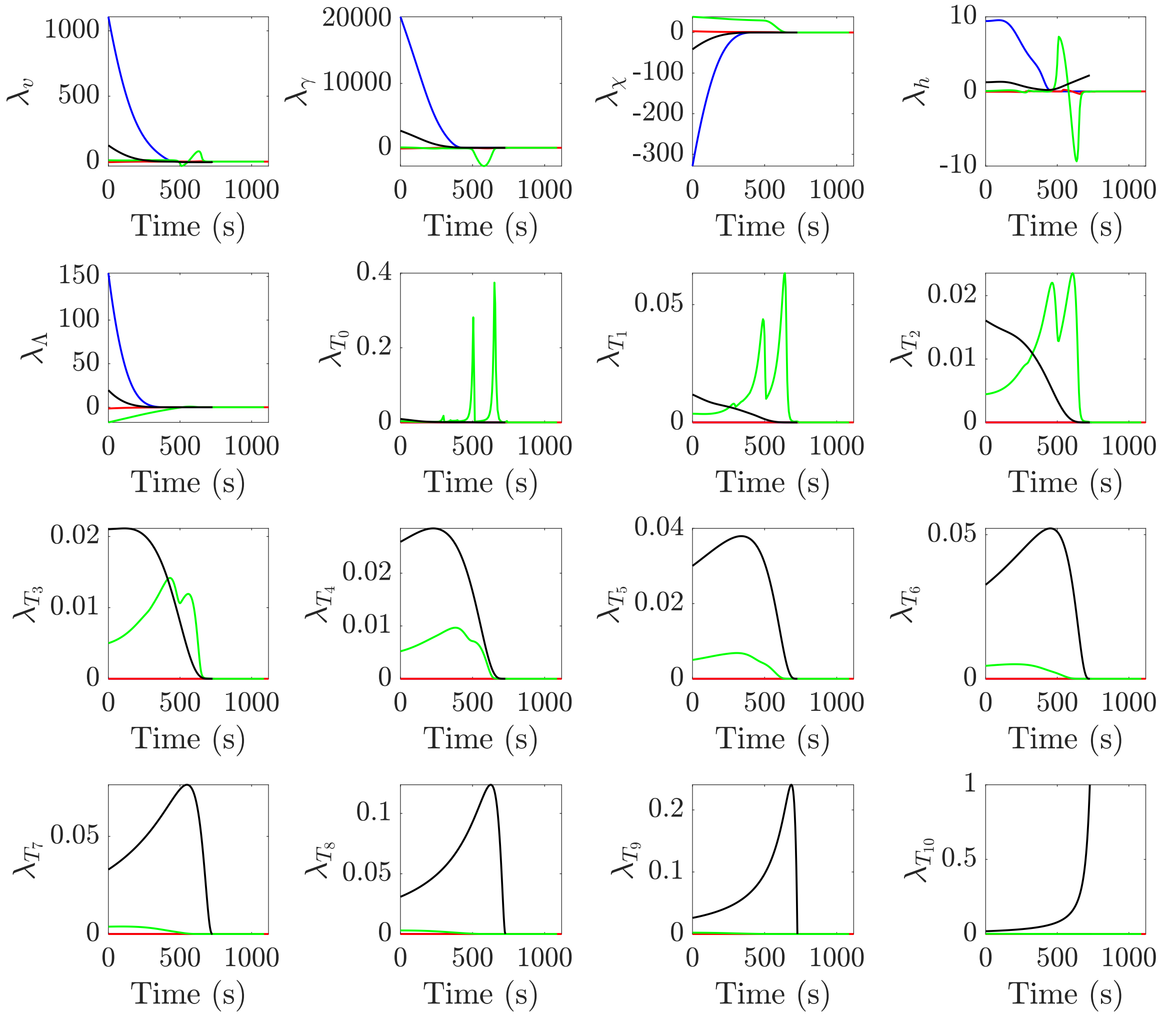

Figure 4.16: Adjoints to the states for Solutions 1-4 to Problem 4.2

5 Full discretization with structure exploitation

In this chapter, we discuss an application of the nonsmooth Newton method with a structure exploitation strategy to optimal control problems involving heat equations. The nonsmooth Newton method (and the majority of nonlinear optimization methods) requires solving a linear system in every iteration, which can become very large in the case of fully discretized optimal control problems, so it becomes crucial to choose a method that can solve these systems efficiently. For this purpose, we analyzed both the structure resulting from the application of full discretization and the substructure resulting from the application of the method of lines to discretize the PDE by representing these matrices in a block-banded way.

Several approaches to solving the linear system are applied to a benchmark quadratic two-dimensional heat equation optimal control problem. The effects of increasing both the time and the spatial nodes on the computational times are used to select the best approaches. The results mentioned so far are included in the paper [100]. Finally, the methods deemed most efficient are applied to a nonlinear version of the benchmark problem with space-dependent controls, and to a problem based on finding an active cooling strategy to control the temperature of the TPS around the stagnation point during re-entry, using an optimal trajectory calculated with the reduced discretization approach in the previous chapter. These applications yield some interesting conclusions on the different advantages of these methods. A discussion on the limitations of this approach to solving optimal control problems and for re-entry trajectory control in particular is included at the end of this chapter.

5.1 Structure exploitation

We chose the nonsmooth Newton method for this exploration; however, any NLP algorithm that involves solving a linear system in every iteration such as SQP or interior point methods would benefit equally from this procedure. For convergence results on the nonsmooth Newton method for fully discretized optimal control problems where controls are only required to be of bounded variation, see [85].

We recall that the nonsmooth Newton method requires solving the system

$$V(Z)\, d = -F(Z), \quad V(Z) \in \partial F(Z) \tag{5.1}$$

in every iteration, where

$$F(Z) = \begin{pmatrix} \nabla_z L(z, \lambda, \mu) \\ h(z) \\ \varphi(-g_1(z), \mu_1) \\ \vdots \\ \varphi(-g_m(z), \mu_m) \end{pmatrix} = 0, \qquad Z = (z, \lambda, \mu) \in \mathbb{R}^{n+m+p}, \tag{5.2}$$

where φ is the Fischer-Burmeister function, the nonlinear problem is defined by f, g and h as in (2.36)-(2.38), and L is its associated Lagrange function.

The Lagrange function associated to Problem 2.8, using a piecewise-linear control approximation and the trapezoidal rule as in (2.34) and adding directly (2.35) if present in the original (OCP), is given by

$$\begin{aligned} L(x, u, \lambda, \mu, \sigma_0, \sigma_f) := \phi(x_0, x_N) + \frac{h}{2}(f_0(x_0, u_0) + f_0(x_N, u_N)) + h \sum_{i=1}^{N-1} f_0(x_i, u_i) \\ + \sum_{i=0}^{N} \mu_i^\top c(x_i, u_i) + \sigma_0^\top \psi_0(x_0) + \sigma_f^\top \psi_f(x_N) \\ + \sum_{i=0}^{N-1} \lambda_i^\top (x_{i+1} - x_i - \frac{h}{2}(f(x_i, u_i) + f(x_{i+1}, u_{i+1}))), \end{aligned}$$

with multipliers $\lambda := (\lambda_0, ..., \lambda_{N-1})^\top \in \mathbb{R}^{n_x N}$, $\mu := (\mu_0, ..., \mu_N)^\top \in \mathbb{R}^{n_c N}$, $\sigma_0 \in \mathbb{R}^{n_0}$ and

$\sigma_f \in \mathbb{R}^{n_f}$. The first order necessary KKT conditions read as

$$\begin{aligned}
\nabla_{\{x,u\}} L(x, u, \lambda, \mu, \sigma_0, \sigma_f) &= 0 \\
x_{i+1} - x_i - \frac{h}{2}(f(x_i, u_i) + f(x_{i+1}, u_{i+1})) &= 0, \quad i = 0, ..., N-1, \\
\psi_0(x_0) &= 0, \\
\psi_f(x_N) &= 0, \\
c(x_i, u_i) \leq 0,\ \mu_i \geq 0,\ \mu_i^\top c(x_i, u_i) &= 0, \quad i = 0, ..., N,
\end{aligned}$$

Applying the Fischer-Burmeister function and rearranging conveniently the equations and the order of the variables, we obtain the nonsmooth equation:

$$F(\bar{Z}) := \left(\begin{array}{c}
\psi_0(x_0) \\
\nabla_{\{x_0,u_0\}} L(x, u, \lambda, \mu, \sigma) \\
\varphi(-c(x_0, u_0), \mu_0) \\
x_1 - x_0 - \frac{h}{2}(f(x_0, u_0) + f(x_1, u_1)) \\
\hline
\nabla_{\{x_1,u_1\}} L(x, u, \lambda, \mu, \sigma) \\
\varphi(-c(x_1, u_1), \mu_1) \\
x_2 - x_1 - \frac{h}{2}(f(x_1, u_1) + f(x_2, u_2)) \\
\hline
\\
\vdots \\
\\
\hline
\nabla_{\{x_N,u_N\}} L(x, u, \lambda, \mu, \sigma) \\
\varphi(-c(x_N, u_N), \mu_N) \\
\psi_f(x_N)
\end{array} \right) = 0, \tag{5.3}$$

where

$$\bar{Z} = (\sigma_0, x_0, u_0, \lambda_0, \mu_0, x_1, u_1, \lambda_1, \mu_1, ..., x_N, u_N, \mu_N, \sigma_f)^\top$$

and the Fischer-Burmeister function φ is applied component-wise. The generalized Jacobians of $F(\bar{Z})$ read as

$$\partial F(\bar{Z}) \subseteq \begin{pmatrix}
\Gamma_0 & \Omega_0 & & \\
\Omega_0^\top & \Gamma_1 & \ddots & \\
& \ddots & \ddots & \Omega_{N-1} \\
& & \Omega_{N-1}^\top & \Gamma_N
\end{pmatrix} \tag{5.4}$$

when defining the matrices

$$\Gamma_k := \begin{pmatrix} H_k & C_k^\top & F_k^\top \\ -S_k C_k & T_k & 0 \\ F_k & 0 & 0 \end{pmatrix}, \qquad k = 1, \dots, N-1,$$

$$\Gamma_0 := \begin{pmatrix} 0 & \Psi_0 & 0 & 0 \\ \Psi_0^\top & H_0 & C_0^\top & F_0^\top \\ 0 & -S_0 C_0 & T_0 & 0 \\ 0 & F_0 & 0 & 0 \end{pmatrix}, \qquad \Gamma_N := \begin{pmatrix} H_N & C_N^\top & \Psi_N^\top \\ -S_N C_N & T_N & 0 \\ \Psi_N & 0 & 0 \end{pmatrix},$$

and

$$\Omega_k := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ G_k & 0 & 0 \end{pmatrix}, \qquad k = 1, \dots, N, \qquad \Omega_0 := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ G_0 & 0 & 0 \end{pmatrix}$$

where, abbreviating $L := L(x, u, \lambda, \mu, \sigma_0, \sigma_f)$,

$$\begin{aligned} H_k &:= \begin{pmatrix} \nabla_{x_k x_k} L & \nabla_{x_k u_k} L \\ \nabla_{u_k x_k} L & \nabla_{u_k u_k} L \end{pmatrix}, \\ C_k &:= \begin{pmatrix} c'_x(x_k, u_k) & c'_u(x_k, u_k) \end{pmatrix}, \\ (S_k, T_k) &\in \partial\varphi(-c(x_k, u_k), \mu_k), \qquad && k = 0, \dots, N, \end{aligned}$$

$$\begin{aligned} F_k &:= \begin{pmatrix} -I_{n_x} - \frac{h}{2} f'_x(x_k, u_k) & -\frac{h}{2} f'_u(x_k, u_k) \end{pmatrix}, \\ G_k &:= \begin{pmatrix} I_{n_x} - \frac{h}{2} f'_x(x_{k+1}, u_{k+1}) & -\frac{h}{2} f'_u(x_{k+1}, u_{k+1}) \end{pmatrix}, \qquad && k = 0, ..., N-1 \end{aligned}$$

$$\Psi_0 := \begin{pmatrix} \psi'_0(x_0) & 0 \end{pmatrix}, \quad \Psi_N := \begin{pmatrix} \psi'_f(x_N) & 0 \end{pmatrix}.$$

The rearrangement of variables and equations we performed yields a nice structure for the block-banded matrix (5.4). Furthermore, its bandwidth and sparsity are mostly determined by the structure of the matrices F_k that represent the derivatives of the dynamics $f(x, u)$. Therefore, in the case these also exhibit a banded or sparse behavior, this structure is particularly advantageous.

The linear system (5.1) solved in every iteration has the form

$$Md = \zeta,$$

where M is a large, block-banded and sparse matrix when N is large. One could factorize the matrix directly with a linear solver that takes advantage of its banded structure or sparsity (note that M is nonsymmetric); however, a block factorization can be performed priorly to exploit the substructure of the blocks. We can define

$$M = UL = \begin{pmatrix} I & \Omega_0 L_1^{-1} & & \\ & \ddots & \ddots & \\ & & I & \Omega_{N-1} L_N^{-1} \\ & & & I \end{pmatrix} \begin{pmatrix} L_0 & & & \\ \Omega_0^\top & L_1 & & \\ & \ddots & \ddots & \\ & & \Omega_{N-1}^\top & L_N \end{pmatrix}$$

with $L_N = \Gamma_N$ and

$$L_j = \Gamma_j - \Omega_j L_{j+1}^{-1} \Omega_j^\top, \quad j = N-1, \ldots, 0. \tag{5.5}$$

Thus, solving a linear equation of type $Md = \zeta$ can be done using forward-backward substitution as follows:

(i) Solve $Uy = \zeta$:

$$\begin{aligned} y_N &= \zeta_N, \\ y_j &= \zeta_j - \Omega_j L_{j+1}^{-1} y_{j+1}, \qquad j = N-1, ..., 0 \end{aligned}$$

(ii) Solve $Ld = y$:

$$\begin{aligned} d_0 &= L_0^{-1} y_0, \\ d_j &= L_j^{-1} \left(y_j - \Omega_{j-1}^\top d_{j-1} \right), \qquad j = 1, ..., N. \end{aligned}$$

Therefore, performing this prior factorization and solving the system by forward-backward substitution reduces the effort of solving the system to the factorization of the L_j matrices, much smaller than M. Additionally, we can take advantage of the possibly banded or sparse substructure of the block matrices that make up M (i.e. in the case of discretized PDEs) by using structure-exploiting linear solvers to factorize each L_j.

The reason why an UL factorization is preferred to a classic LU one in this case, being U an upper triangular matrix and L a lower triangular matrix, is due to possible rank defficiencies in the submatrices in the presence of initial conditions. An analogue development attempting to obtain an LU factorization would require the matrix Γ_0

to be inverted as a first step in (5.5), and this matrix is bound to be rank defficient in this case: the first and last block rows in Γ_0 are

$$\begin{pmatrix} 0 & \Psi_0 & 0 & 0 \\ 0 & F_0 & 0 & 0 \end{pmatrix},$$

which despite $\Psi_0 \in \mathbb{R}^{n_0 \times (n_x+n_u)}$ and $F_0 \in \mathbb{R}^{n_x \times (n_x+n_u)}$ having full rank, leads to Γ_0 having these two linearly dependent block rows and being singular. Therefore, performing an UL factorization which requires inverting Γ_N as a first step is a more convenient option. The rank of Γ_N also depends on the problem and the submatrices, so it is not guaranteed to be invertible either, but it is under some assumptions (see Corollary 2.2): for example, if the Hessian H_N is positive definite (which can be enforced by adding γI to H_N, being $\gamma > 0$ a small but large enough constant), the matrix C_N has full rank, and there are no final boundary conditions, i.e. $n_f = 0$. This is the case for the PDE problems explored in the next sections. From this point forward, when an LU factorization is mentioned, we refer to the UL factorization described here.

5.2 Software implementation

In order to implement this methodology and explore its possibilities, a software package was implemented in C++ with two separate modules:

- An explicit derivative generator for optimal control problems using the free computer algebra system GiNaC [9] to perform symbolic differentiation
- An optimization solver that implements the nonsmooth Newton method with a linesearch globalization (Algorithm 2.5) and exploits the block-banded structure from (5.4).

The discretization module `derivGenerator` works as an interface to GiNaC that generates a `.cpp` file in which all the functions from the optimal control problem and their derivatives are written explicitly as methods of a class called `OPCO`. GiNaC uses symbolic differentiation to provide explicit derivatives by application of the chain rule and algebraic operations to the functions in the problem represented as symbolic expressions. Note that this method is limited to functions that can be represented in terms of elementary functions and algebraic operations, so it cannot be applied when piecewise-defined functions are present, for example. The code writes first derivatives of the objective function, dynamics and constraints needed to calculate $\nabla_{\{x,u\}}L$ and submatrices F_k, G_k, C_k, S_k and T_k, as well as their second derivatives in the form of a method that calculates directly the Hessians H_k. Symbolic differentiation comes with the advantage of accuracy provided by the exact explicit derivatives over other numerical methods, such as finite differences. However, problems may arise since full rank is not guaranteed for the Hessians, unlike with BFGS updates (see Section 2.1.1), and it becomes crucial that the user provides a well-defined problem.

The main optimization module consists of a main class `NLProblem` with the following members: an object of the class `OPCO`, that provides all the functions and derivatives from the optimal control problem, a series of vectors that store all the aforementioned submatrices, and the number of time nodes N and time step $h = 1/N$. The method `solveNLPNewton` provides a solution to the fully discretized optimal control problem using the nonsmooth Newton method for a certain initial point (z_0, λ_0). Notice that it is not necessary to formulate the fully discretized problem, but only to call the methods in `OPCO` to define and update the submatrices and the function $F(\bar{Z})$ as formulated in (5.3) in every iteration.

All the matrices are stored in a sparse matrix format defined in the class `SpMatrix`, where only the row and column indices of each element and their values are stored in three separate vectors. This means a great saving in memory since instead of the full number of elements $m \times n$, only $3 \times n_z$ elements are stored, where n_z is the number of non-zero elements in the matrix. Some operations such as addition or multiplication of matrices can become very costly on the other hand, but these were not needed in our algorithms. It was necessary to implement other methods that in fact imply fewer operations with the sparse format, such as transposition or multiplication by a vector. Some sparse linear solvers also require for the matrix to be converted to Compressed Colum (CC) format, which can save even more memory as it requires storing even fewer elements, but makes it more complex to manipulate matrices.

A high-level diagram of these main classes in the optimization module, their members and methods, and the external components is depicted in Figure 5.1.

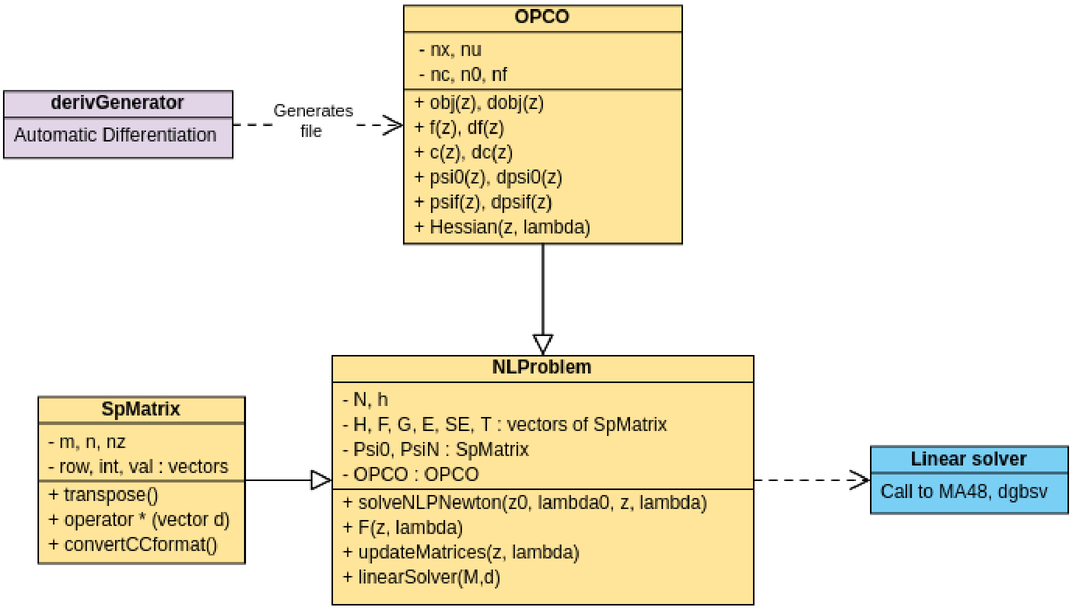

Figure 5.1: Diagram of nonlinear optimization software

We selected two linear solvers that exploit the structure of linear system (5.1): MA48 [31] from the HSL Mathematical Software Library for sparse nonsymmetric systems, and subroutine `dgbsv` from the LAPACK library [5] for banded matrices. The purpose of the test is to compare four different approaches, namely:

(L1) Solving the linear system directly with `dgbsv`.

(L2) Solving the linear system directly with MA48.

(L3) Solving the linear system by forward-backward substitution after block LU factorization, using `dgbsv` to solve the subsystems.

(L4) Solving the linear system by forward-backward substitution after block LU factorization, using MA48 to solve the subsystems.

The first task that arises is to compare these approaches and decide which one performs more efficiently in terms of computational time, which is done in the next section on a quadratic benchmark problem. However, further tests have revealed that there might be different advantages and disadvantages for each approach, depending on the problem at hand. From the point of view of implementation, it is worth mentioning that:

- The sparse solver MA48 allows for the matrices to be provided in sparse format, while `dgbsv` requires a transformation into a banded format (all the elements from the diagonals are stored in a vector), which can result in a larger memory requirement if the number of diagonals is not small.
- The LU decomposition requires for a larger number of smaller subsystems to be solved forward and backward. Ideally, these could be solved in parallel, but it would require a more complicated implementation.

All the following tests have been performed on an Intel Core i7-8565U, 4 × 1.80 GHz processor.

5.3 Quadratic PDE benchmark problem

In order to test our implemented full discretization approach and nonsmooth Newton method, and to compare the described structure and substructure-exploiting approaches to solving the linear system (5.1) in every iteration, we formulated the following benchmark quadratic problem adapted from [11] involving a two-dimensional heat equation with a constant controlled temperature at the edge $u(t)$ over a squared surface $[0, H] \times [0, H]$ (see Figure 5.2):

Problem 5.1. *Minimize*

$$\int_{t_0}^{t_f} \int_0^H \int_0^H (T(x, y, t) - T_a)^2 \ dx \ dy \ dt + \int_{t_0}^{t_f} \gamma u(t)^2 \ dt$$

w.r.t. $u(t)$*, subject to the constraints*

$$\begin{aligned}
&\frac{\partial T}{\partial t} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) \\
&T(x,0,t) - \lambda \frac{\partial T}{\partial y}(x,0,t) = u(t), && \text{(C1)} \\
&T(0,y,t) - \lambda \frac{\partial T}{\partial x}(0,y,t) = u(t), && \text{(C2)} \\
&\frac{\partial T}{\partial x}(H,y,t) = 0, && \text{(C3)} \\
&\frac{\partial T}{\partial y}(x,H,t) = 0, && \text{(C4)} \\
&T(x,y,0) = T_0, \\
&0 \le T(x,y,t) \le T_{max}, \\
&0 \le u(t) \le u_{max},
\end{aligned}$$

for all $t \in [t_0, t_f]$*,* $(x,y) \in [0,H] \times [0,H]$*.*

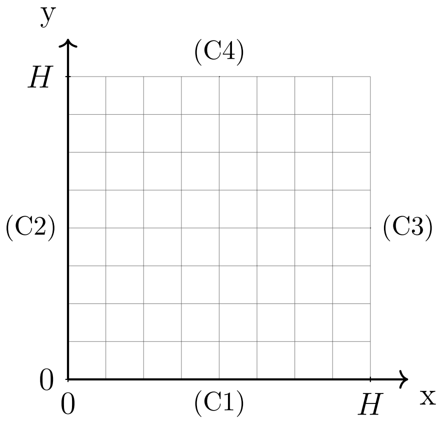

Figure 5.2: Spatial domain for Problem 5.1 with boundary constraints (C1)-(C4)

The goal is for the temperature to be as close as possible to the constant temperature T_a over the whole domain and time interval, and a control regularization term with a small constant $\gamma > 0$ is added to the objective function. This can help to guarantee the full rank of the Hessians. We apply the method of lines from Section 2.3.3 to transform this PDE optimal control problem into an ODE one: We perform a spatial finite-difference discretization based on an uniform grid over the spatial domain $[0, H] \times$

$[0, H]$:

$$
\begin{aligned}
x_i &= i\Delta x, \quad i = 0, 1, ..., m_x, \quad \Delta x = H/m_x, \\
y_j &= j\Delta y, \quad j = 0, 1, ..., m_y, \quad \Delta y = H/m_y,
\end{aligned}
$$

and defining $T_{ij}(t) = T(x_i, y_j, t)$, we obtain an approximation of the PDE as the system of ODEs

$$
\begin{aligned}
\frac{\partial T_{ij}}{\partial t}(t) = \frac{\alpha}{\Delta x^2}(T_{i,j-1} - 2T_{ij} + T_{i,j+1}) + \frac{\alpha}{\Delta y^2}(T_{i-1,j} - 2T_{ij} + T_{i+1,j}), \\
i = 0, ..., m_x, \; j = 0, ..., m_y,
\end{aligned}
$$

completed by using the discretized boundary conditions to approximate the values outside of the domain using central differences as in (2.53):

$$
\begin{aligned}
T_{i,m_y+1} &= T_{i,m_y-1}, \qquad T_{i,-1} = \frac{2\Delta x}{\lambda}(u(t) - T_{i0}) + T_{i1}, \qquad i = 0, ..., m_x, \\
T_{m_x+1,j} &= T_{m_x-1,j}, \qquad T_{-1,j} = \frac{2\Delta y}{\lambda}(u(t) - T_{0j}) + T_{1j}, \qquad j = 0, ..., m_y.
\end{aligned}
$$

With this discretization, the objective function becomes

$$
\int_{t_0}^{t_f} \sum_{i=0}^{m_x} \sum_{j=0}^{m_y} w_i v_j (T_{ij}(t) - T_a)^2 + \gamma u(t)^2 \, dt,
$$

where

$$
w_i = \Delta x \begin{cases} 0.5, & \text{if } i = 0 \text{ or } i = m_x \\ 1, & \text{otherwise,} \end{cases} \qquad v_j = \Delta y \begin{cases} 0.5, & \text{if } j = 0 \text{ or } j = m_y \\ 1, & \text{otherwise.} \end{cases}
$$

The structure and substructure of the linear problem following the discretization and rearrangement procedures presented in Section 5.1 is depicted in Figure 5.3 for a small spatial grid size of 4 × 4 and 10 time nodes, for better visualization. As observed in Section 5.1, the structure of the matrix V_k is banded and sparse, due to the full discretization approach. The substructure of each $\Gamma_i, i = 0, ..., N$ that composes V_k is banded and sparse as well, due to the use of the method of lines to discretize the PDE as presented in (2.61).

For our test example, we chose $T_a = 0.2$, $\alpha = 1.0$, $\lambda = 0.5$, $t_0 = 0$, $t_f = 2$, $H = 1.0$, $T_0 = 0$, $T_{max} = 0.7$, $\gamma = 10^{-8}$ and $u_{max} = 1.0$. We tested the four approaches to

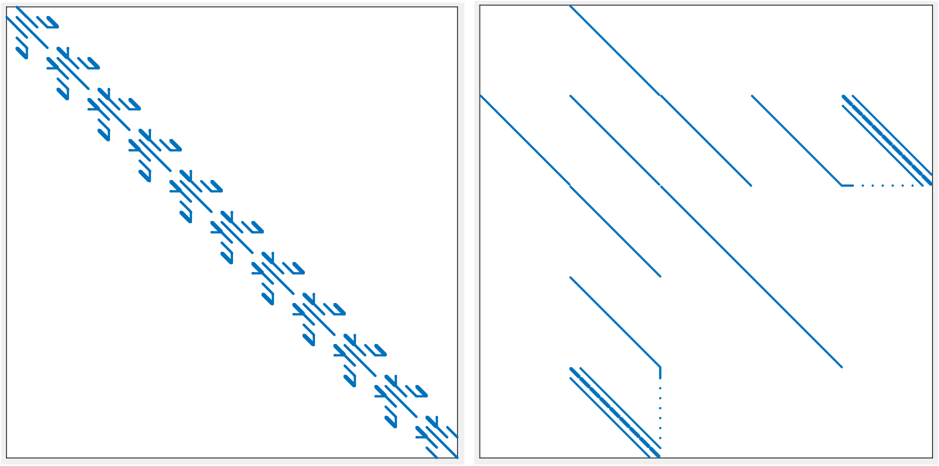

Figure 5.3: Matrix and submatrix structure for Problem 5.1: V_k and Γ_0 with $m_x = 8$, $m_y = 8$, $N = 10$.

solving the linear system in Section 5.2, (L1)-(L4), with different cases of spatial grid points ($m_x \times m_y$) and time nodes (N). The feasibility and optimality tolerances were set to 10^{-8}, and all methods produced the same solution in the same number of global iterations in each case.

The computational results are shown in Table 5.1, as well as the linear system size (LS size) of the system (5.1) that needs to be solved in every iteration that varied from around 5000 to almost 50000.

Despite the size of the linear systems being of the same order, increasing spatial grid points yields a more complex matrix structure than increasing time grid points and therefore, requires a higher effort. This is visible by comparing the times for each row in the table. Furthermore, it seems likely that MA48 might not be the best choice for solving the problem directly, but comparing approaches (L2) and (L4), it is clear that the forward-backward substitution method offers a vast improvement when using MA48.

In the case of the subroutine `dgbsv` from LAPACK, approach (L3) performed slightly worse than approach (L1). This is probably due to LAPACK already being able to deal with big, complex matrices very efficiently, compensating for the slight extra effort of performing the forward-backward substitution as in approach (L3). In conclusion, approaches (L1) and (L4) were the best performers, and the obtained computational

$m_x \times m_y$	N	LS size	Total CPU time			
			(L1)	(L2)	(L3)	(L4)
4 × 4	50	5253	1.13 s	7.66 s	1.48 s	1.50 s
8 × 8	50	16677	10.35 s	311.19 s	12.51 s	8.38 s
12 × 12	50	34629	195.43 s	4875.85 s	232.60 s	125.49 s
8 × 8	50	16677	10.35 s	311.19 s	12.51 s	8.38 s
8 × 8	100	33027	49.94 s	1484.30 s	49.51 s	36.17 s
8 × 8	150	49377	45.66 s	3488.94 s	55.19 s	37.28 s
$m_x \times m_y$	N	LS size	Time/iteration			
			(L1)	(L2)	(L3)	(L4)
4 × 4	50	5253	0.07 s	0.51 s	0.10 s	0.10 s
8 × 8	50	16677	0.86 s	25.93 s	1.04 s	0.70 s
12 × 12	50	34629	8.14 s	211.99 s	9.69 s	5.23 s
8 × 8	50	16677	0.86 s	25.93 s	1.04 s	0.70 s
8 × 8	100	33027	2.77 s	82.46 s	2.75 s	2.01 s
8 × 8	150	49377	3.26 s	249.21 s	3.94 s	2.66 s

Table 5.1: Computational results for Problem 5.1

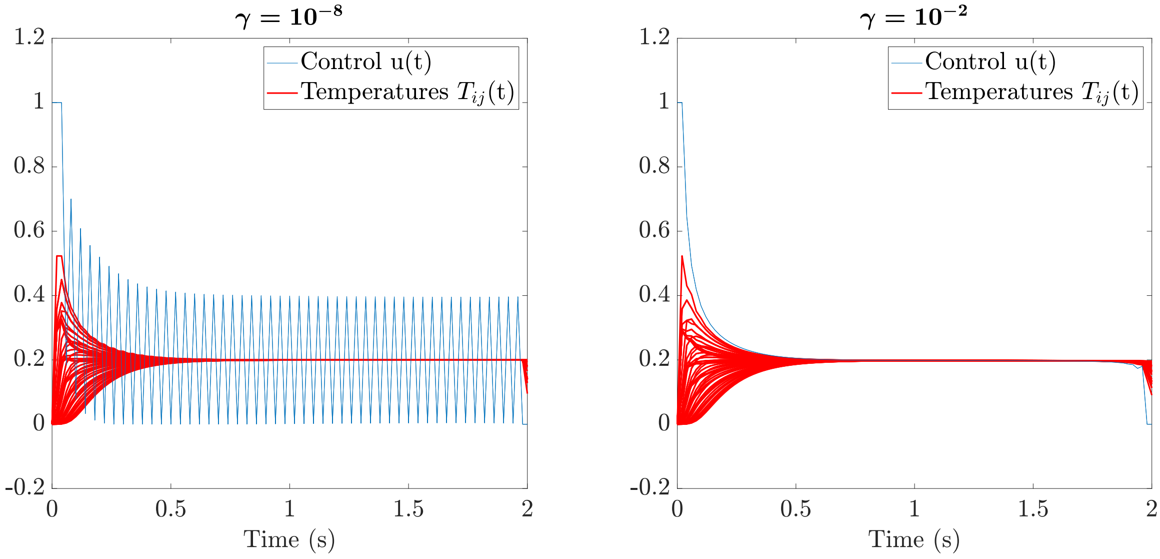

Figure 5.4: Solutions to Problem 5.1 for $m_x = 8$, $m_y = 8$, $N = 100$

times seem to increase at a slower pace with these two approaches than with the rest when increasing the numbers of spatial and time nodes.

The solution is depicted in Figure 5.4 for $m_x = 8$, $m_y = 8$ and $N = 100$ and two regularization parameters: $\gamma = 10^{-8}$ and $\gamma = 10^{-2}$. Following the disussion in ([41], Section 7.1.1), when a control appears linearly in an optimal control problem, we can either get a bang-bang solution (i.e. the control switches from upper to lower bound) or a singular control. This behavior is defined by the switching function, defined for (OCP) as

$$\Gamma(x, \lambda) = \lambda^\top f'_u(x, u). \tag{5.6}$$

Note that if u appears linearly in $f(x, u)$, then $f'_u(x, u)$ depends only on x. In that case, we have that the optimal control u^* follows

$$u^*(t) = \begin{cases} u_{min} & \text{if } \Gamma(x^*(t), \lambda^*(t)) > 0, \\ u_{max} & \text{if } \Gamma(x^*(t), \lambda^*(t)) < 0, \\ \text{undefined} & \text{if } \Gamma(x^*(t), \lambda^*(t)) > 0, \end{cases} \tag{5.7}$$

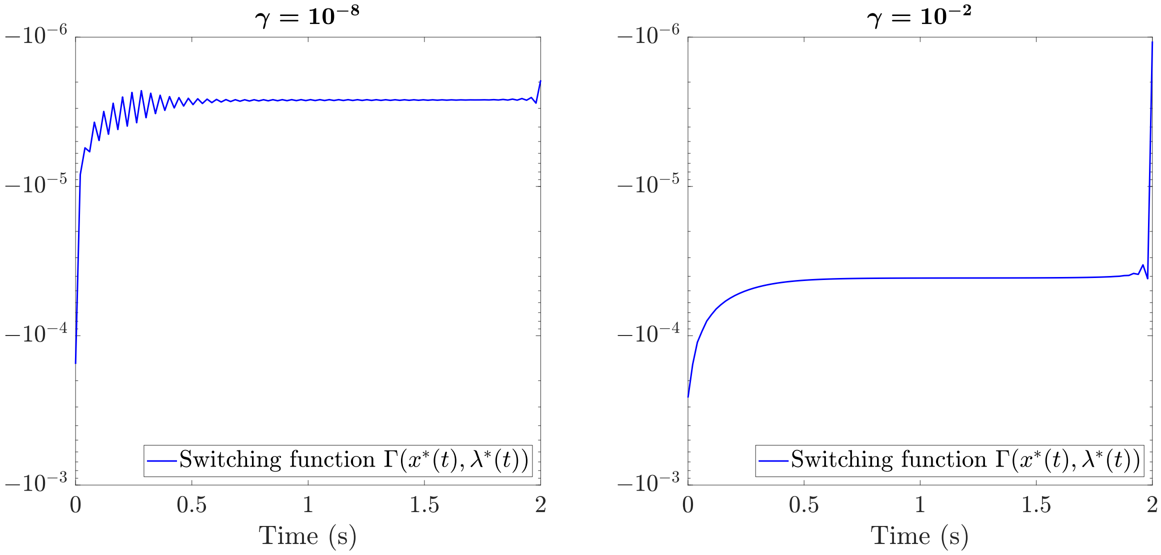

Figure 5.5: Switching functions for Problem 5.1 for $m_x = 8$, $m_y = 8$, $N = 100$

The obtained switching functions, plotted in Figure 5.5, suggest a bang-singular-bang structure when $\gamma = 0$, where the control is at its upper bound at the first time node and at its lower bound at the last one, with a singular arc in between. While the middle values of the switching function are not exactly at 0 by a small margin between 10^{-6}

and 10^{-4}, due most likely to small numerical errors, it is clear that they are centered very close to 0, and the lowest or most negative value is at the first node while the highest is at the last one. Therefore, the oscillations in the solution when γ is close to 0 can be expected since there is no convergence analysis on what happens in singular arcs, although there are techniques to better deal with this type of solutions [134]. This serves as well to highlight the importance of the Hessians being positive definite for the performance of the optimizer: with $\gamma = 10^{-8}$, the matrix is quasi-singular, which produces a much more irregular solution than in the case of $\gamma = 10^{-2}$. This can be appreciated as well in the profile of the switching functions for each solution.

The multipliers associated to the discretized dynamic constraints for each state, $\lambda_i(t)$ for $i = 0, ..., m_x \cdot m_y$, are depicted in Figure 5.6. Since the states and control bound constraints are inactive almost everywhere in the solution, their associated multipliers μ are constantly 0 almost everywhere due to the complementarity condition (2.7). These are the discrete multipliers as described by the discrete local minimum principle in Chapter 2 and interpreted thereafter as discrete versions of their continuous counterparts in (OCP).

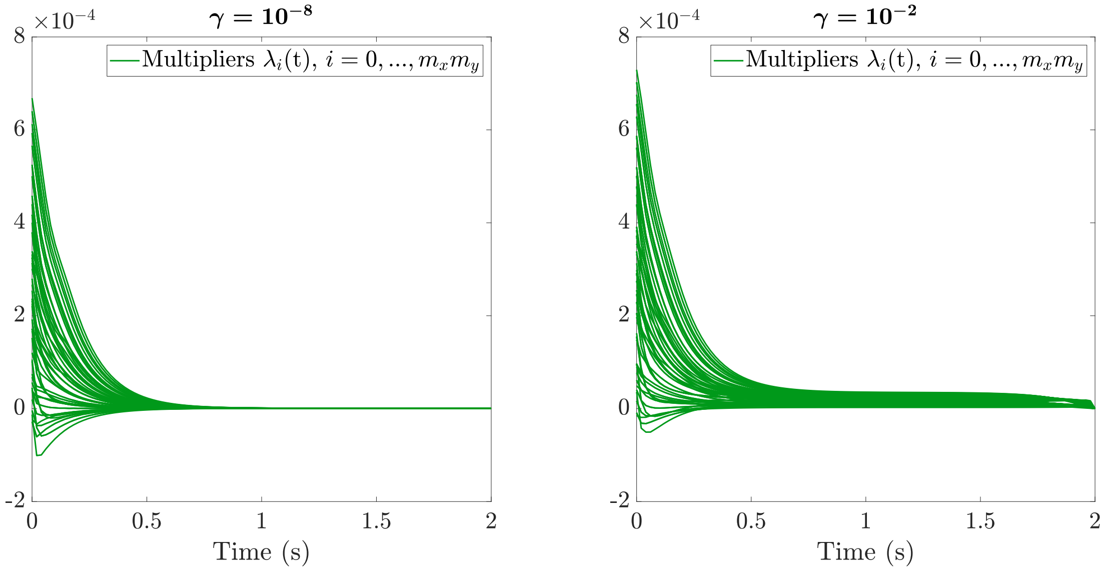

Figure 5.6: Multipliers for Problem 5.1 for $m_x = 8$, $m_y = 8$, $N = 100$

5.4 Nonlinear PDE problem

We increase the complexity of the problem by considering the following nonlinear optimal control problem from [47, 107] with a slight variation for the controls. The goal is to obtain a certain temperature trajectory over a given rectangular domain $[0, x_{max}] \times [0, y_{max}]$. The PDE includes now a non-linear source term $S(T)$ given by

$$S(T) = S_{max}e^{-\beta_1/(\beta_2+T)},$$

for some specified $S_{max}, \beta_1, \beta_2 \geq 0$. The objective is for the temperature to follow a specified trajectory $\tau(t),\ t \in [t_0, t_f]$, in a sub-domain $\Omega = [x_0, x_{max}] \times [y_0, y_{max}]$, where $0 < x_0 < x_{max}$, $0 < y_0 < y_{max}$. The controls are now different for each sides of the domain and depend on the spatial variables as well: $u_1(t, x)$ and $u_2(t, y)$. The optimal control problem is formulated as

Problem 5.2. *Minimize*

$$\begin{aligned}
&\int_{t_0}^{t_f} \int_{\Omega} (T(x, y, t_f) - \tau(t))^2 \; d(x, y) \; dt + \\
&\int_{t_0}^{t_f} \int_{0}^{x_{max}} \gamma u_1(t, x)^2 \; dxdt + \int_{t_0}^{t_f} \int_{0}^{y_{max}} \gamma u_2(t, y)^2 \; dydt
\end{aligned}$$

w.r.t. $u_1(t, x), u_2(t, y)$, *subject to the constraints*

$$\begin{aligned}
&\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + S(T), \\
&T(x, 0, t) - \lambda \frac{\partial T}{\partial y}(x, 0, t) = u_1(t, x), \\
&T(0, y, t) - \lambda \frac{\partial T}{\partial x}(0, y, t) = u_2(t, y), \\
&\frac{\partial T}{\partial x}(x_{max}, y, t) = 0, \\
&\frac{\partial T}{\partial y}(x, y_{max}, t) = 0, \\
&T(x, y, 0) = T_0, \\
&0 \leq T(x, y, t) \leq T_{max}, \\
&0 \leq u_1(t, x) \leq u_{max}, \\
&0 \leq u_2(t, y) \leq u_{max},
\end{aligned}$$

for all $t \in [t_0, t_f]$, $(x, y) \in [x_0, x_{max}] \times [y_0, y_{max}]$.

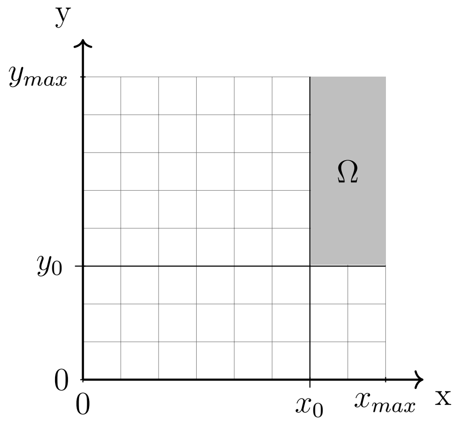

Figure 5.7: Spatial domain for Problem 5.2 with sub-domain Ω

The same space discretization methodology applied to the previous problem can be applied here, including the source term $S(T)$ in the ODE system and discretizing the controls $u_1(t,x), u_2(t,y)$ along their spatial dimension as well, generating a series of controls discretized in space:

$$\begin{aligned} u_{1i}(t) &:= u_1(t,x_i), \quad i = 0, ..., m_x, \\ u_{2j}(t) &:= u_2(t,y_j), \quad j = 0, ..., m_y. \end{aligned}$$

Note that the number of control functions as they would appear in the ODE system discretized in space is $(m_x+1)+(m_y+1)$, and it increases with every increase in the number of spatial nodes.

The discretized boundary conditions are reformulated again to obtain the values outside of the boundary as follows:

$$\begin{aligned} T_{i,-1} &= \frac{2\Delta x}{\lambda}(u_{1i}(t) - T_{i0}) + T_{i1}, \qquad i = 0, ..., m_x, \\ T_{-1,j} &= \frac{2\Delta y}{\lambda}(u_{2j}(t) - T_{0j}) + T_{1j}, \qquad j = 0, ..., m_y. \end{aligned}$$

For this test, we use the constants $\lambda = 0.5$, $\alpha = 1.0$, $S_{max} = 0.5$, $\beta_1 = 0.2$, $\beta_2 = 0.05$, $T_0 = 0$, $T_{max} = 0.7$, $\gamma = 10^{-3}$, $u_{max} = 1.0$, and the boundaries for the domains $t_0 = 0$, $t_f = 2$, $x_{max} = 0.8$, $y_{max} = 1.6$, $x_0 = 0.6$, $y_0 = 0.6$. The desired temperature trajectory is defined by

$$\tau(t) = \begin{cases} 0, & 0 < t \leq 0.2 \\ 1.25(t - 0.2), & 0.2 < t \leq 0.6; \\ 0.5, & 0.6 < t \leq 1.0; \\ 0.5 - 0.75(t - 1.0), & 1.0 < t \leq 1.4; \\ 0.2, & 1.4 < t \leq 2.0. \end{cases}$$

Case	$m_x \times m_y$	N	Number of states n_x	Number of controls n_u	LS size
1	8 × 8	50	81	18	19278
2	12 × 12	50	169	26	38454
3	16 × 16	50	289	34	64158
4	20 × 20	50	441	42	96390
5	8 × 8	100	81	18	38178
6	8 × 8	200	81	18	75978
7	8 × 8	300	81	18	113778
8	8 × 8	400	81	18	151578

Table 5.2: Problem size for test cases on discretized Problem 5.2

We used approaches (L1) and (L4) from the last subsection, since they proved to be the most efficient for the quadratic problem, and a feasibility and optimality tolerance of 10^{-6}. Note that due to the nonlinearity of the problem, it is required to provide first derivative and Hessian values in every iteration to define the linear system (5.1) in the nonsmooth Newton method (in the quadratic case, the derivatives remain constant), which are provided by the symbolic differentiation module.

We experimented once again with several cases for the number of space and time discretization nodes, this time trying to test the capacity of the solvers even further. Table 5.2 shows an overview of the sizes of resulting optimal control problems after space discretization, as well as the sizes of resulting NLPs after time discretization. It is worth mentioning that while the grid sizes might not seem too large, going up to 20 × 20, the number of the optimal control problem states and controls increases to values that would pose a challenge to any standard solver.

Case	$m_x \times m_y$	N	CPU time approach (L1)	CPU time approach (L4)	Objective function	Iterations
1	8×8	50	16.71 s	7.81 s	7.41×10^{-4}	11
2	12×12	50	120.65 s	57.80 s	8.99×10^{-4}	11
3	16×16	50	542.05 s	220.18 s	1.18×10^{-3}	10
4	20×20	50	1602.94 s	790.79 s	1.38×10^{-3}	10
5	8×8	100	38.02 s	19.63 s	7.38×10^{-4}	12
6	8×8	200	80.45 s	42.40 s	7.28×10^{-4}	13
7	8×8	300	137.35 s	69.72 s	7.27×10^{-4}	14
8	8×8	400	182.61 s	95.32 s	7.27×10^{-4}	14

Table 5.3: Computational results for Problem 5.2

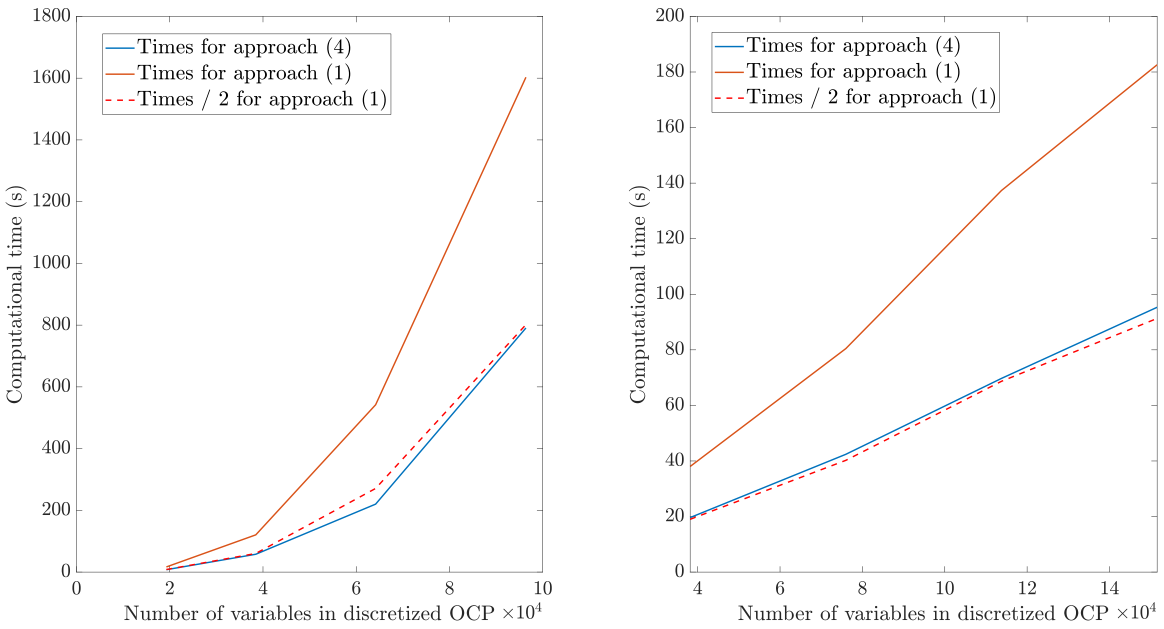

Figure 5.8: Computational times for Cases 1-4 and for Cases 5-8

Computational results and objective function values can be found in Table 5.3. Once again, the effect of increasing the grid size in Cases 1-4 yields higher CPU times than increasing the number of time points in Cases 5-8. This effect is clearly depicted in Figure 5.8: when incrementing the number of spatial nodes, the computational times grow in a superlinear manner, whereas when incrementing the number of time nodes, they grow linearly. It is also interesting to notice that the computational times

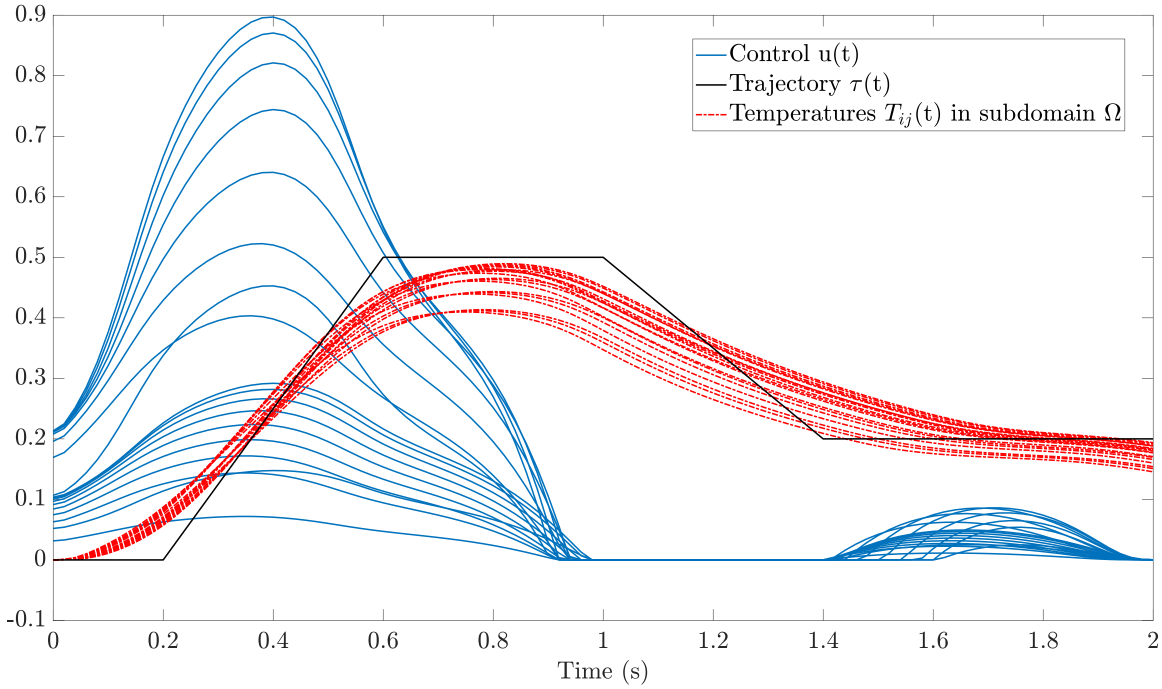

Figure 5.9: Solution for Case 5: $m_x = 8$, $m_y = 8$, $N = 100$

using approach (L1) were around twice as large as using approach (L4), which is why the halved times for approach (L1) are also plotted for comparison. Furthermore, the performance in terms of objective function values is also slightly affected by the increases in the spatial nodes, which is not the case when the time nodes are increased, see again Table 5.3.

For reference and comparison, in [47, 107] a simpler version of this problem with a single control is solved with a modified multiple shooting method. For a grid size of 16 × 16 as in Case 3 and 60 time nodes, this method required 6283 seconds of computation time. While one-on-one comparisons cannot be made due to other variables that can affect the CPU times, the 220 seconds that approach (L4) took for Case 3 and given the added complexity of the problem considered here gives further proof that full discretization with structure exploitation offers a vast advantage in efficiency compared to reduced discretization approaches with solvers for small and dense problems.

The control and the temperature obtained at the nodes in Ω against the target trajectory $\tau(t)$ can be seen in Figure 5.9 for Case 5. Following the notation from Section

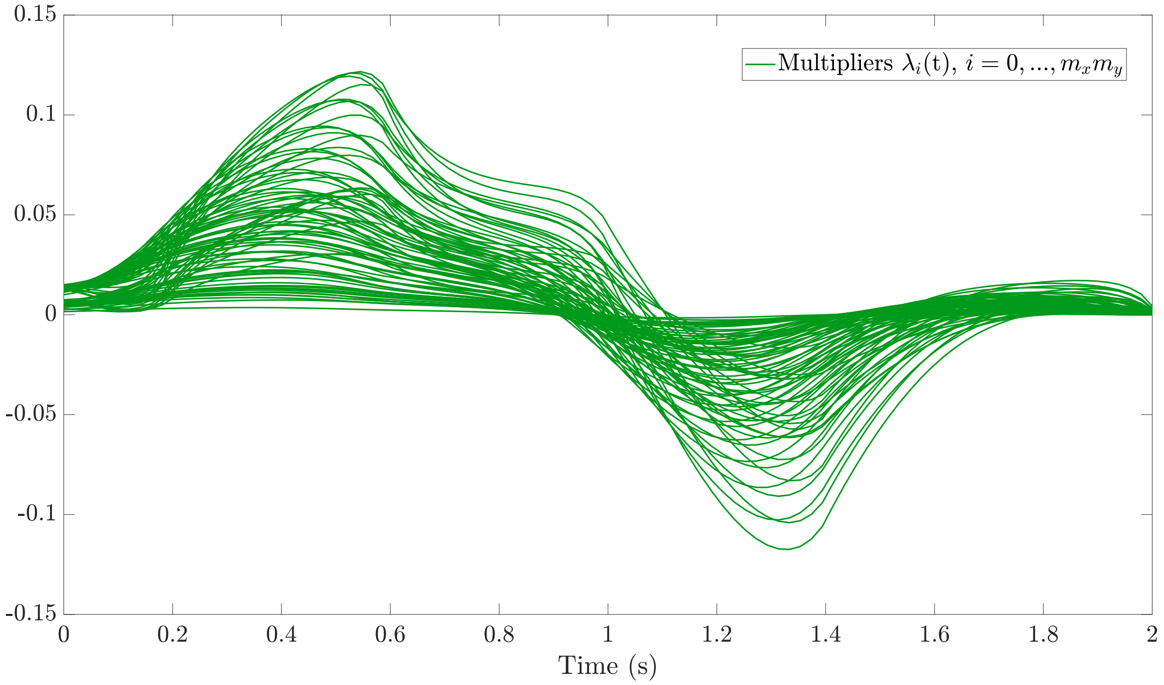

Figure 5.10: Multipliers for dynamic constraints for Case 5: $m_x = 8$, $m_y = 8$, $N = 100$

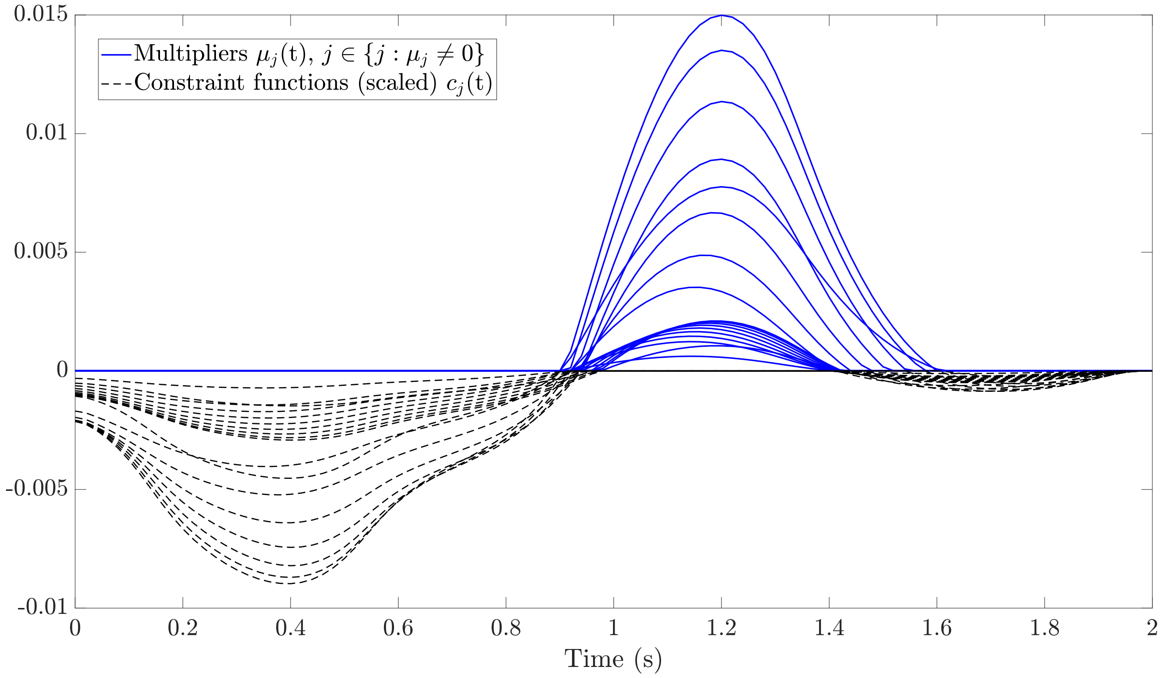

Figure 5.11: Multipliers for box constraints for Case 5: $m_x = 8$, $m_y = 8$, $N = 100$

5.1, the multipliers associated to the discretized dynamic constraints for each state, $\lambda_i(t)$ for $i = 0, ..., m_x m_y$, are depicted in Figure 5.10. As for the box constraints on the temperature and the control, $0 \leq T_{ij}(t) \leq T_{max}$ for $i = 0, ..., m_x$, $j = 0, ..., m_y$, and $0 \leq u(t) \leq u_{max}$, they are mostly inactive except for one interval in which the controls are at their lower bound 0. Given that the multipliers $\mu_i(t)$ for $i = 1, ..., 2(m_x m_y + 1)$ are constantly at 0 wherever the constraints are inactive due to the complementarity conditions, we depict only the non-zero multipliers in Figure 5.11 and their associated constraint functions, the only ones that do become active at some point: $-u_j(t) \leq 0,\ j = 1, ..., n_u$, scaled for a better visualization of the complementarity between the functions and the multipliers.

From this test, we can conclude that our implemented methodology is capable of dealing with nonlinear problems and a large number of controls without a significant deterioration in the computational times for both approaches (L1) and (L4). We can also observe that an increase in the number of spatial nodes can result in a computational challenge for these algorithms, but an increase of the number of time nodes does not present major changes in their performance or efficiency.

5.5 Re-entry temperature control problem

As previously mentioned, an accurate representation of the heating over the surface of a vehicle during re-entry requires CFD simulations or experimental data. We can however produce an artificial model to obtain a possible heating evolution by extending the heating values obtained for an optimal trajectory in the previous chapter along the surface of a re-entry capsule. The purpose here is to test the efficiency of our method in a possible re-entry scenario with a PDE optimal control problem. Hence, we consider the two-dimensional heat equation from (3.21) for the heat transfer through a transversal laminar section of the TPS from the stagnation point:

$$\rho_{TPS}\ c_{p,TPS} \frac{\partial T(t,x,y)}{\partial t} = \lambda_{TPS} \left(\frac{\partial^2 T}{\partial x^2}(t,x,y) + \frac{\partial^2 T}{\partial y^2}(t,x,y) \right), \tag{5.8}$$

for $(t, x, y) \in [0, t_f] \times [0, D_x] \times [0, D_y]$. Boundary conditions are defined as in (3.23)-(3.26), with $T_{cool}(t, x)$ as control. Due to the challenges that a coupled re-entry trajectory problem posed for this method, which are discussed in the next section, we

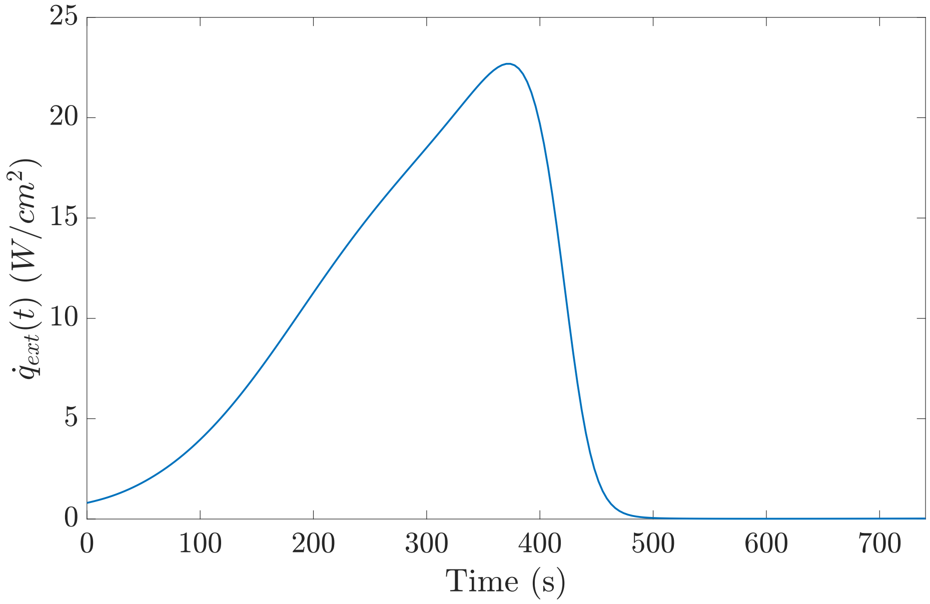

Figure 5.12: Input external heating $\dot{q}_{ext}(t)$ from optimal trajectory in Section 4.3

used the external heating values from the optimal trajectory calculated for Solution 1 of Section 4.3 directly as an input to the heat equation, that is, from (3.11) and (3.15):

$$\dot{q}_{ext}(t,x) := (1-rx)k_E\sqrt{\frac{\rho(h(t))}{R_N}}v(t)^3 + \epsilon\sigma T_{air}(h(t))^4, \tag{5.9}$$

where $h(t), v(t)$ and t_f are the ones obtained for said optimal trajectory, see Figure 5.12 for a representation of $\dot{q}_{ext}(t)$.

A schematic representation of the discretized domain and the external and internal cooling heat flux is depicted in Figure 5.13, with $\dot{q}_{ext}(t,x)$ as defined in (5.9) and $\dot{q}_{cool}(t,x)$ from (3.27).

We follow now the method of lines through the development of [27, 136] to discretize the equation and include a radiation term based on the Stefan-Boltzmann law (3.15) to model the vertical radiative propagation of heat through the layers of the TPS. With the same spatial grid as in the previous problem, we express the discretized

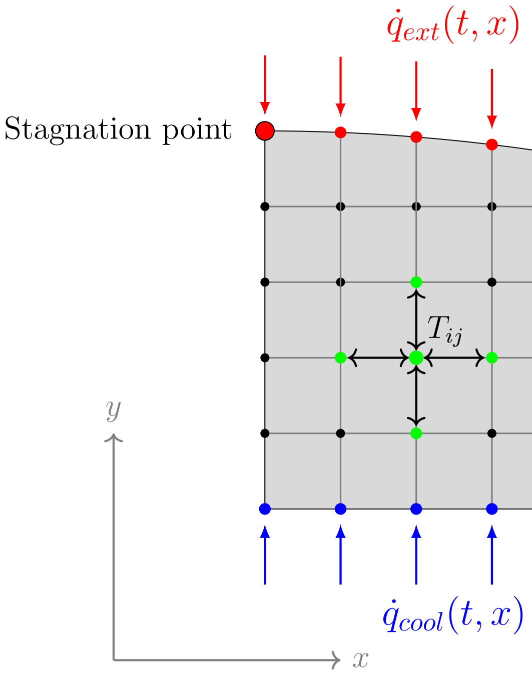

Figure 5.13: 2D section from the stagnation point with external and cooling heat flux

derivatives as:

$$q_{ij}^{x}(t) = \lambda_{TPS}\left(\frac{T_{i-1,j}(t)-T_{ij}(t)}{\delta_x}\right), \tag{5.10}$$

$$q_{ij}^{y}(t) = \lambda_{TPS}\left(\frac{T_{i,j-1}(t)-T_{ij}(t)}{\delta_x}\right) + \epsilon\sigma(T_{i,j-1}(t)^4 - T_{ij}(t)^4), \tag{5.11}$$

$$i = 0,...,m_x,\ j = 1,...,m_y - 1.$$

Taking into account the boundary conditions (3.23) and (3.24), where the controls are involved in $\dot{q}_{cool}$ (see (3.27)), we have

$$q_{i0}^{y}(t) = \dot{q}_{ext}(t,x_i) - \epsilon\sigma T_{i0}^4, \tag{5.12}$$

$$q_{i,m_y}^{y}(t) = \dot{q}_{cool}(t,x_i) - \epsilon\sigma T_{iM}(t)^4, \quad i = 0,...,m_x \tag{5.13}$$

and from (3.26),

$$T_{-1,j} = T_{1,j}, \quad T_{m_x+1,j} = T_{N-1,j}, \quad j = 0,...,m_y \tag{5.14}$$

Hence, we obtain as ODE system:

$$\rho_{TPS}\ c_{p,TPS}\frac{\partial T_{ij}}{\partial t}(t) = \frac{1}{\delta_x}\left(q_{ij}^{x}(t) - q_{i+1,j}^{x}(t)\right) + \frac{1}{\delta_y}\left(q_{ij}^{y}(t) - q_{i,j+1}^{y}(t)\right)$$

$$i = 0,...,m_x,\ j = 0,...,m_y. \tag{5.15}$$

Note that the addition of the radiative heat flux in (5.11) makes these dynamics nonlinear for all $T_{ij}(t)$, increasing the complexity of the problem. We approximate as well the controls with the discretization $T_{cool}(t, x_i) \approx u_i(t),\ i = 0, ..., m_x$, which yields a collection of $m_x + 1$ controls. As explained in Section 3.4, the goal is to maintain the coolant temperature $u_i(t)$ as high as possible, which is equivalent to minimizing the sum of all $(u_i(t) - T_{cool,max})^2$. Apart from lower and upper bounds on the controls, we require that the temperature at the inner nodes T_{i,m_y} is within certain boundaries $T_{in,min}$ and $T_{in,max}$. We only require this at the inner nodes since the external heating values are already prescribed by the optimal trajectory we obtained in Section 4.3. Its high values make it hard to control the temperature of the whole surface, except for the inner nodes that are closer to the cooling system. Taking all these elements into account, the semi-discretized optimal control problem is formulated as

Problem 5.3. *Minimize*

$$\int_{t_0}^{t_f} \sum_{i=0}^{m_x} (u_i(t) - T_{cool,max})^2 \ dt$$

w.r.t. $u_i(t),\ i = 0, ..., m_x$, *subject to the constraints*

$$\begin{aligned}
&\rho_{TPS}\ c_{p,TPS} \frac{\partial T_{ij}}{\partial t}(t) = \frac{1}{\delta_x}\left(q^x_{ij}(t) - q^x_{i+1,j}(t)\right) + \frac{1}{\delta_y}\left(q^y_{ij}(t) - q^y_{i,j+1}(t)\right),\\
&T_{in,min} \leq T_{i,m_y}(t) \leq T_{in,max},\\
&T_{cool,min} \leq u_i(t) \leq T_{cool,max},
\end{aligned}$$

for all $t \in [0, t_f],\ i = 0, ..., m_x,,\ j = 0, ..., m_y$.

We summarize the constants and chosen parameters for this test in Table 5.4. All constants related to heat transfer are obtained from [80, 136], once again with the intention of reproducing the case of the Apollo capsule using realistic values. The remaining constants and quantities involved in (5.9) are as defined for the calculation of $\dot{q}_{ext}(t, x)$ in Section 4.3.

We chose $m_x = m_y = 10$, which produces a grid of 121 space nodes, and $N = 200$ as number of time nodes, which results in a discretized nonlinear problem with 26532 variables and a linear system size of 59697 in the application of the nonsmooth Newton method. Note that these choices are among the largest within the cases studied in the previous section. The feasibility and optimality tolerance were both set to 10^{-6}.

Constant	Value	Constant	Value
ρ_{TPS}	528.6 kg/m^3	t_f	740.54 s
$c_{p,TPS}$	2742.35 J / kg K	D_x	0.8 m
$\lambda_{p,TPS}$	0.242 W / K m	D_y	0.07 m
α_q	35 J / m^2 s K	$T_{in,min}$	200 K
ϵ	0.8	$T_{in,max}$	400 K
σ	5.67 $\cdot 10^{-8}$ J / m^2 s K^4	$T_{cool,min}$	14 K
r	0.03	$T_{cool,max}$	300 K

Table 5.4: Constants and parameters for Problem 5.3

Both approaches (L1) and (L4) were used again to solve the problem, and, despite (L4) being the front-runner in terms of efficiency so far, in this case approach (L1) proved to be a better option. The reason is that this problem has a lower number of constraints since only the temperature at the inner nodes is bounded, unlike in the previous test problems where all the temperature functions were constrained. Hence, we have a substructure with a much lower number of subdiagonals for this case, which is a big advantage for the LAPACK subroutine for banded matrices that is used in approach (L1). This results in approach (L1) being able to solve the large linear system in every iteration in around 3 seconds, much faster than approach (L4) which needed around 10 seconds.

Besides the big difference in efficiency, some complications were encountered with approach (L4) involving the convergence of the method. Due to the larger number of time nodes and the higher complexity of this problem, approach (L4) failed to find a solution and despite managing to reach a fairly low value of the merit function, it was not able to reach the value of the optimality tolerance. This is likely due to the higher numerical errors produced by this approach as opposed to solving the system directly. With the forward-backward substitution, any numerical error is dragged and amplified when solving every subsystem, and this effect is particularly worsened when the number of subsystems is large, i.e. when the number of time nodes N is large. Therefore, in terms of achieving convergence, approach (L1) can also be a better option in some cases.

For all of these reasons, we can extract the insight that the best approach both in terms of convergence and efficiency depends on the problem at hand. As our goal was to validate our method, test our approaches and try to solve a re-entry heating problem with this methodology, experimentation with more problems is out of the scope of this work. However, there should be interesting outcomes from the application of these approaches to different problems.

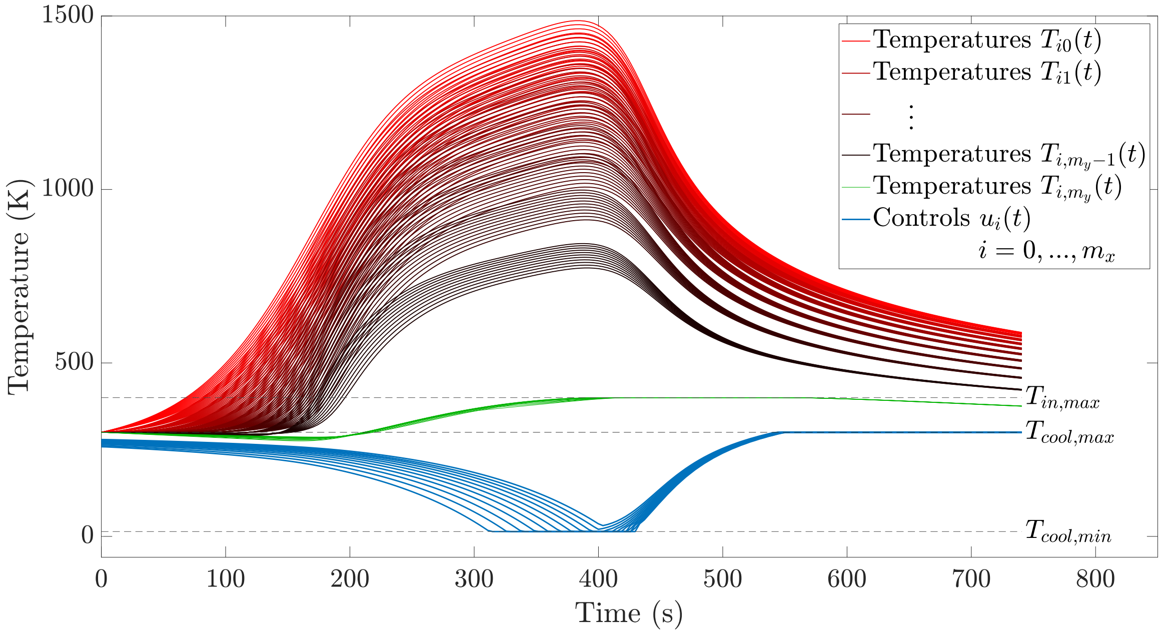

Figure 5.14: Solution of Problem 5.3

The problem was solved by approach (L1) in 270.79 s, after 69 iterations. The solution is depicted in Figure 5.14: the temperatures for the different layers $j = 0, ..., m_y - 1$ are depicted in a red-to-black gradient, the constrained inner temperatures T_{i,m_y} in green, and the controls in blue. The bounds on the controls and the upper bound on inner temperatures T_{i,m_y} are also plotted. Despite the efforts made for a clear representation of the solution, the temperatures are unfortunately not easy to distinguish from one another since their values overlap; however, it can be noticed that there are several distinct layers which represent the temperatures $T_{ij}, i = 0, ..., m_x$ for all nodes at a certain depth y_j in the TPS. Within each layer, it can be noticed that the temperatures vary within a lower and upper bound. The highest temperature evolution in each layer corresponds to the spatial node in that layer closer to the stagnation point, and the lowest temperature to the furthest from the stagnation point, as it could be expected.

We can clearly see that there is a first time arc during which the temperature rises and reaches almost 1500 K in the exterior, and there is a quick decrease thereafter and until the end of the time interval (this reproduces the behavior of the prescribed external heating, see Figure 5.12). During the first time arc, the controls or cooling temperatures are used to lower the temperature of the inner nodes to keep it from rising over the upper boundary $T_{in,max} = 400$ K. It can also be noticed that some of the controls are at their lower bound of 14 K for a longer time period than others: this depends once again on which horizontal node they correspond to and its distance to the stagnation point. The ones closer to the stagnation point suffer a higher heat load and therefore, they need more cooling.

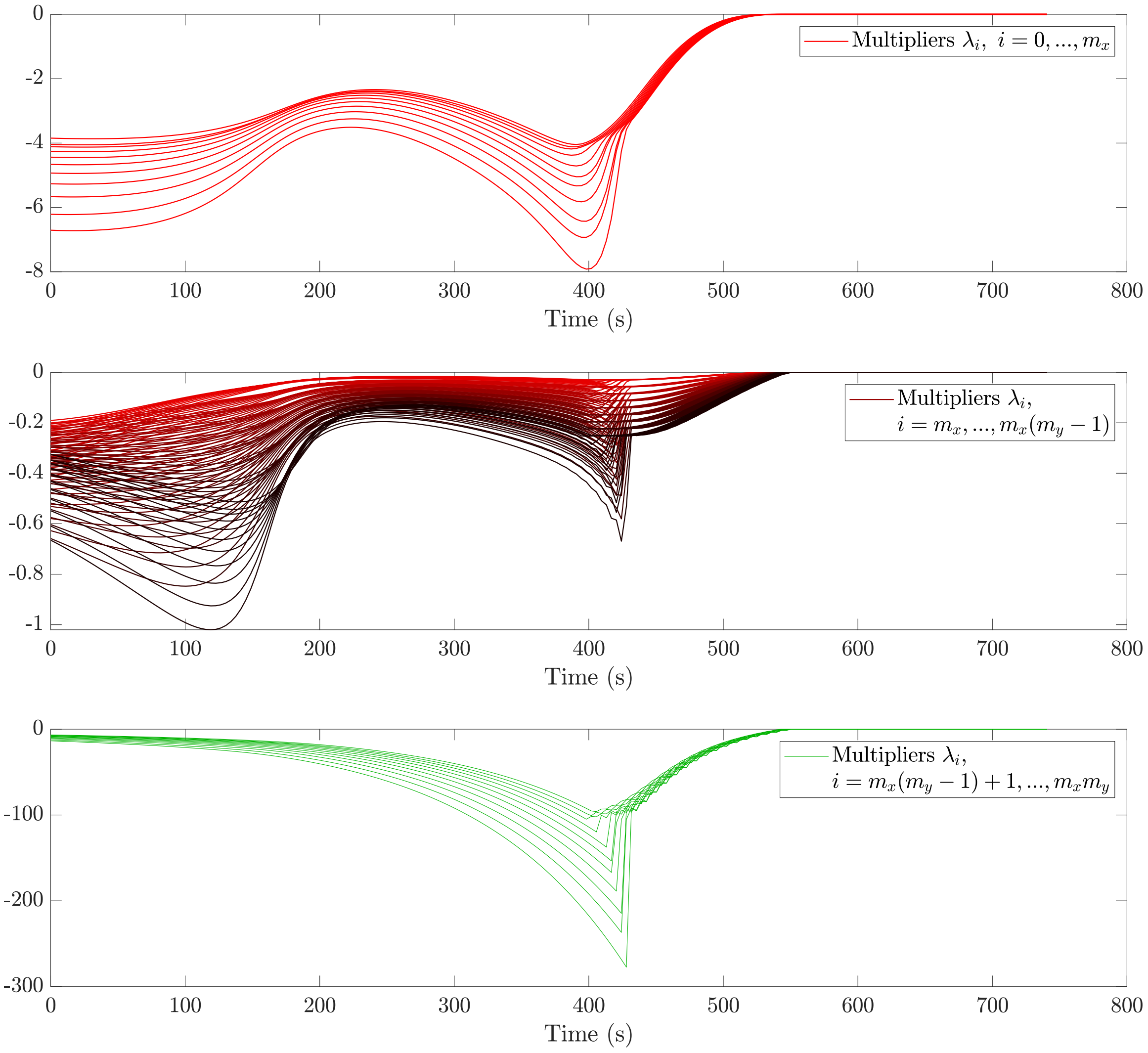

Figure 5.15: Multipliers λ_i for dynamic constraints of the discretized Problem 5.3

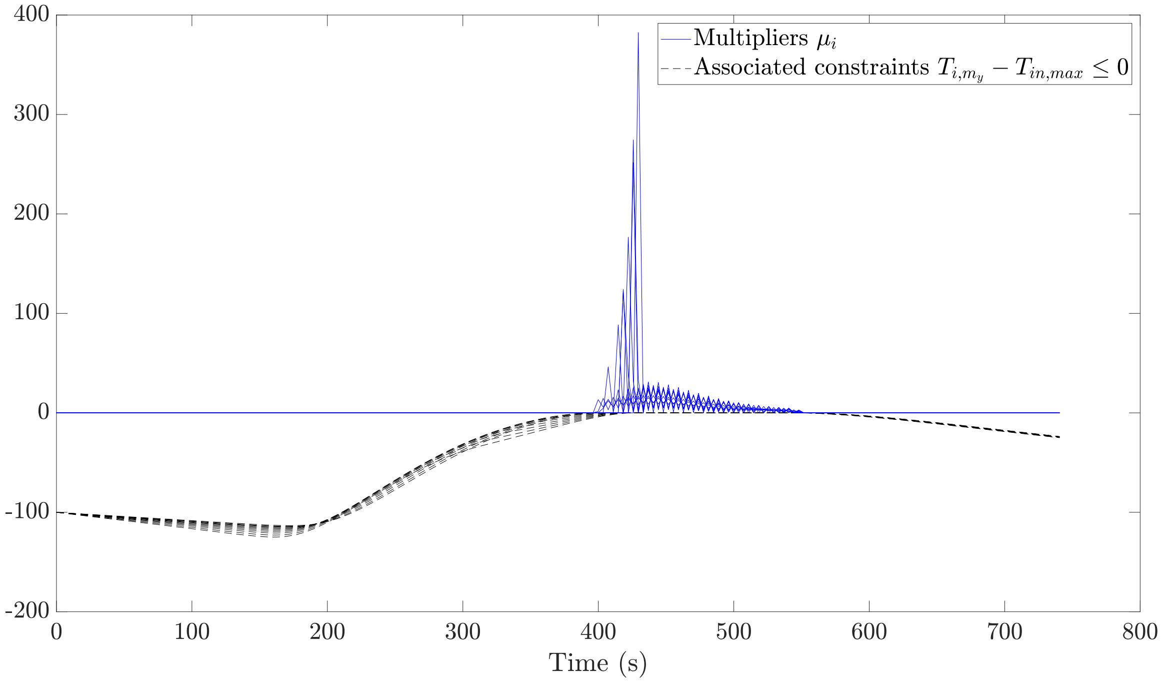

Figure 5.16: Multipliers for state box constraints of the discretized Problem 5.3

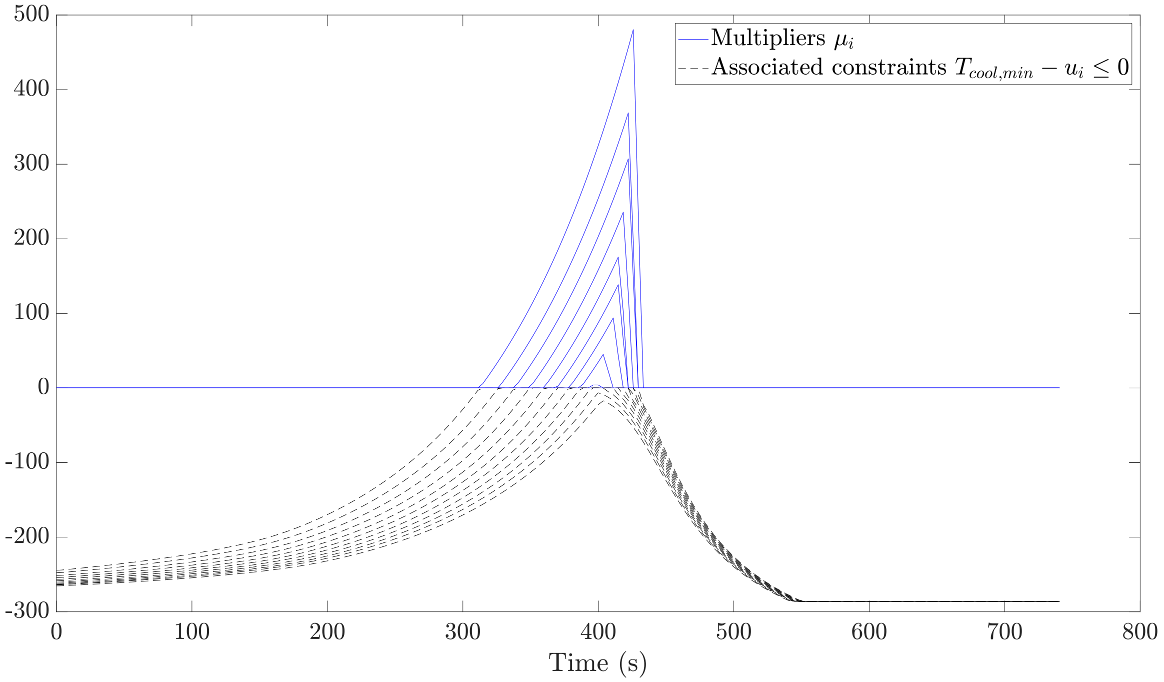

Figure 5.17: Multipliers for control box constraints of the discretized Problem 5.3

It is also worth noticing that in the latter part of the time interval, when the external heating is lower, all the controls return to their maximum boundary, which means that the cooling system does not need to be activated and is therefore "deactivated" by setting the controlled cooling temperatures to their highest bound. In fact, the inner temperatures decrease at the end with no effect from the cooling system. Hence, this solution provides the minimum decrease in the cooling temperatures in order to keep the temperature at the inner nodes below their upper boundary, which corresponds as well to the minimum use of coolant, as explained in Section 3.4.

The discretized Lagrange multipliers $\lambda_i,\ i = 0, ..., m_x m_y$ are depicted in Figure 5.15, separated between the external, middle and interior layers for better visualization (notice the differences in the scales on the y-axis) and following the same color code as in Figure 5.14 for a better association to their corresponding temperature states. The jumps in some of them are likely due to the state constraints $T_{i,m_y}(t) \leq T_{in,max}$ becoming active. As for the multipliers associated to the box constraints μ_i, we depict once again only those that become non-zero at some point: those associated to the box constraints on the states $T_{i,m_y} \leq T_{in,max}$ in Figure 5.16 and on the controls $T_{cool,min} \leq u_i$ in Figure 5.17. Once again, the dynamics of the states are recognizable in their corresponding multipliers and the complementarity conditions are clearly being satisfied by the box constraints and their associated multipliers, as expected from the interpretation of the discrete local minimum principle in Section 2.3.

Finally, we include the values of the merit function $\Theta(Z^k)$ (2.15) and the step-size α_k (see Algorithm 2.5) for each iteration k in Figure 5.18 in order to depict the superlinear convergence of the method established by Theorem 2.4. Despite the large merit function value for the initial guess (around $2 \cdot 10^5$), the method manages to eventually reach a very fast convergence, as can be observed in the last iterations.

5.6 Discussion of results and limitations

It was made clear in the last chapter that a reduced discretization approach with a solver such as OCPID-DAE1 is easily able to solve a coupled ODE-PDE re-entry trajectory problem, as long as not many controls are involved. It was not the case in the problems we considered in Section 4.3, with only two controls for the trajectory

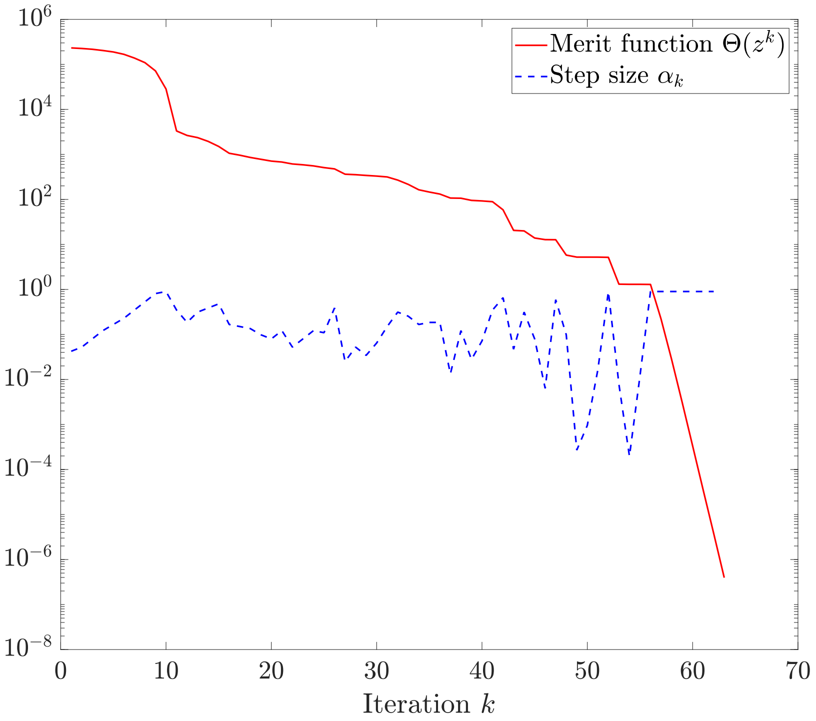

Figure 5.18: Progress of the nonsmooth Newton method on Problem 5.3

governing both the aerodynamics in the ODE system and the temperature variables in the PDE. However, when we consider PDEs in two (or more) dimensions where the controls depend on the spatial variables as well, the number of controls can increase after spatial discretization. Since OCPID-DAE1 uses a linear solver for small, dense problems, it can run into problems when the number of controls is large.

To illustrate this issue, we show in Table 5.5 a comparison between the computational times for OCPID-DAE1 and approaches (L1) and (L4) with some of the smallest nonlinear PDE problem cases from Section 5.4. All the methods ran for the same number of iterations, which was the number of iterations that approaches (L1) and (L4) took to converge for each case, between 10 and 13. Besides the clear disadvantage in the computational times and their rapid growth even when the time nodes are increased for the same problem, OCPID-DAE1 did not manage to converge after 1000 iterations for any of the cases.

$m_x \times m_y$	N	Full discretization			OCPID-DAE1	
		LS size	CPU time (L1)	CPU time (L4)	LS size	CPU time
4×4	50	6630	1.19 s	1.05 s	500	18.60 s
4×4	100	13130	2.32 s	2.02 s	1000	46.23 s
8×8	50	19278	16.71 s	7.81 s	900	171.82 s
8×8	100	38178	38.02 s	19.63 s	1800	436.16 s

Table 5.5: Computational times for Problem 5.1 with approaches (L1), (L4) and OCPID-DAE1

However, it should also be mentioned that our full discretization with a nonsmooth Newton method strategy did not converge either when trying to find solutions to re-entry trajectory problems. The cause of this seems to be the high nonlinearity of these problems and the ill-conditioned matrices they yield in every iteration. In fact, optimal control re-entry problems with heating as a constraint or objective function are considered as examples of ill-conditioned systems to test modified Newton methods in [14, 26]. According to [26, 121], for ill-conditioned problems, the Newton direction and the steepest-descent direction of the merit function (2.15) are almost orthogonal, which leads to very small stepsizes and an extremely slow convergence.

We reproduced this slow and stagnated convergence behavior while trying for solving a simple re-entry trajectory problem with a coupled PDE and minimizing the heating, as the one solved in Section 4.3. We show the behavior through this example, but it is also reproduced in different scenarios and variations of the problem. It ran for 55 iterations and stopped when stepsize α reached a value under the set threshold of 10^{-9}, see in Figure 5.19 the evolution of the merit function $\Theta(Z)$, the stepsize α and the matrix condition number calculated with Matlab:

$$cond(A) = \|A\| \cdot \left\|A^{-1}\right\| \tag{5.16}$$

Some effors to improve the convergence have been made with the introduction of so-called natural level functions in [14], and a modified Newton method that includes a regularization of the merit function (2.15) in [72] as explained in Chapter 2 solving the system

$$(V_k^\top V_k + \nu_k I)d = V_k^\top F(Z^k) \tag{5.17}$$

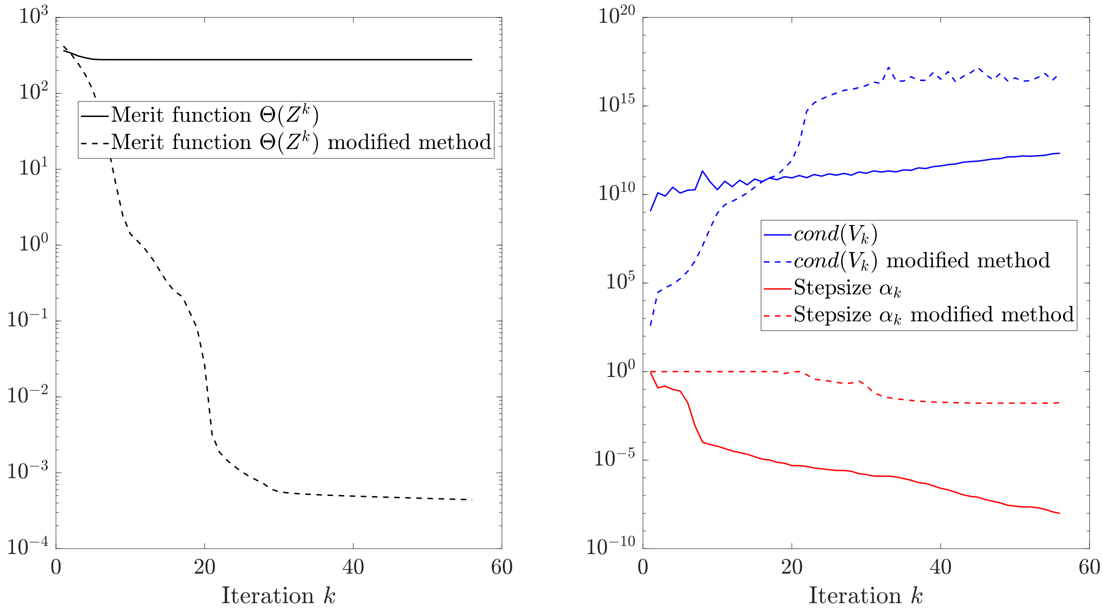

Figure 5.19: Progress of the original and modified nonsmooth Newton method on a fully discretized re-entry problem

instead of (5.1), where $\nu_k = \min\{\Theta(Z^k), \left\|\nabla\Theta(Z^k)\right\|\}$. This new system can be more dense and have more diagonals than the original one, which can considerably affect the performance of the considered linear solvers. It was clear in our test that the computational times increased due to this factor and the need to calculate the product $V_k^\top V_k$ in every iteration.

The progress in the convergence function for this modified Newton method for the first 55 iterations is also depicted in Figure 5.19: there was an initial improvement shown in a rapid decrease of the merit function in the first 20 iterations. However, the progress stagnated afterwards and didn't manage to reach the convergence criteria either after 1000 iterations with the same tendency. This behavior is consistent with the one from the unmodified method given that the parameter ν_k that modifies the linear system depends on the value of the merit function $\Theta(Z^k)$, and both methods are the same when $\nu_k = 0$. Therefore, we can see clearly that when $\Theta(Z^k)$ reaches lower values, the condition number and stepsize have a similar behavior for both methods.

Moreover, from a theoretical point of view, superlinear convergence with this modification (or without it) is not guaranteed by the results in [72] when non-singularity cannot be assumed for the generalized Jacobian $V^* \in \partial F(Z^*)$ at the accumulation point Z^* generated by this method. In Figure 5.19 we can see that the modified method tries to converge to a point with a very high condition number $cond(V^*)$, and the very high condition numbers of $cond(V_k)$ in the last iterations mean that these matrices are nearly singular, which suggests V^* could be singular or very ill-conditioned.

Another challenge is that practical problems are often large, highly nonlinear and ill-conditioned and therefore exhibit a small neighborhood of convergence [121], which leads to the problem of finding a close enough starting point. An integrated, feasible solution was used for these problems, but it seems it was not a good enough initial guess to reach convergence. Due to the complexity of our re-entry trajectory problems, this is not a trivial task and it can be a time consuming extra step, which defeats our purpose of finding efficient methods to solve these problems.

Therefore, we can conclude that a full discretization approach with the nonsmooth Newton method is not the best option for re-entry trajectory problems, but it can be useful and efficient to solve large-scale and discretized PDE optimal control problems in the context of re-entry. Since OCPID-DAE1 can fail when trying to solve these problems but it can find solutions for re-entry trajectory problems robustly and efficiently, these two approaches complement each other well.

6 Conclusion and outlook

This thesis studies different discretization methods and nonlinear optimization algorithms for optimal control problems, including the implementation of a structure exploitation methodology for fully discretized optimal control problems, with an application to minimizing heating during atmospheric re-entry.

First, we presented an overview of the basic concepts and results of nonlinear optimization, and expanded on two types of algorithms to solve general nonlinear problems: SQP methods and nonsmooth Newton methods. Local and global algorithms and results on both local and global convergence were recalled. As one of the major drawbacks in the numerical application of these methods, the solvability of the linear system yielded by each of the methods was discussed and some sufficient conditions were presented. Then, we focused on optimal control theory, presenting the necessary conditions of the local minimum principle for general ODE optimal control problems with mixed control-state constraints, and discussing the numerous approaches to solving optimal control problems. The direct discretization approaches that are the basis of the metodologies applied in this thesis are presented in detail. In particular, a version of the discrete local minimum principle presented in [41] for general one-step methods is described for the full discretization method. Finally, the method of lines to obtain semidiscretized PDEs is explained in detail, and the sparsity of the yielded ODE systems after discretization is depicted.

Before applying the described methodologies, a compilation of all the models required to pose different re-entry problems was described in Chapter 3. Some approximations to the US Standard Atmosphere of 1976 were considered to obtain air temperature and pressure values, needed to calculate the aerodynamic forces and external heating. Lift and drag were calculated with two different models based on the vehicle considered for each scenario: one for the German Sänger concept, and one for the Apollo capsule. Due to the infeasibility of using high accuracy simulations to calculate external heating, the Sutton-Graves formula was used to calculate convective heat flux, the Stefan-Boltzmann law was used to calculate radiative heat flux, and the temperature of the TPS was calculated with a standard heat equation. An active cooling system was also considered for the problem of minimizing coolant usage under certain temperature constraints, as defined in Chapter 5. Finally, a parametric shape model for the Apollo capsule was described, with the intent of analyzing the effect of shape parameters on trajectory optimization.

A first attempt towards obtaining optimal re-entry trajectories with minimum heating was made in Chapter 4 using OCPID-DAE1. The software proved to be able to obtain optimal trajectories robustly and efficiently for different models, scenarios and conditions, to optimize shape parameters, and to solve coupled ODE-PDE problems for different objective functions with the aim of minimizing both convective flux and TPS temperature. The obtained optimal trajectories are consistent with the different posed problems, and it was possible to find solutions in the cases where the heat equation was fully coupled and involved in the optimization process.

Then, a structure exploitation strategy for fully discretized optimal control problems with the nonsmooth Newton method and the details of its implementation were presented in Chapter 5. A convenient rearrangement of the variables in the fully discretized problem using the trapezoidal rule lead to a block-banded, sparse matrix structure that can be exploited in the linear system that needs to be solved in every iteration. Furthermore, a block-UL factorization reduces the computational effort by iteratively solving much smaller subsystems, and allows for the structure of the submatrices to be exploited as well. This methodology was implemented as a C++ software including an explicit derivative generator for optimal control problems and using sparse matrix format to store matrices for additional memory saving.

A benchmark quadratic PDE problem was used to expose the large-scale, sparse and banded structure and substructure of such problems, and to compare the computational times required by different strategies combined with different solvers that exploit banded and sparse structures. The tests show that using a solver for banded matrices from LAPACK was much more efficient than using the sparse solver MA48 when solving the whole matrix. However, exploiting the substructure of the problem with the block factorization combined with MA48 gave the best results and radically reduced the computational time from solving the entire system directly with MA48.

Numerical and computational results were presented for a more complex, nonlinear version of the same discretized PDE problem with an increased number of controls. Both the structure-exploiting approach with LAPACK and the substructure-exploiting approach with MA48 succeeded in finding solutions for increasing time and space grid points within reasonable computational times, which grew linearly when increasing the number of time nodes and superlinearly when increasing the number of space nodes. An optimal active cooling problem during re-entry using the external heating values from the optimal trajectory calculated by OCPID-DAE1 was successfully solved with the first approach, while the second one ran into problems trying to achieve convergence, most likely due to accumulated numerical errors amplified by the forward-backward substitution. Considering the complexity and large size of the resulting discretized problem, the solver managed to find an optimal cooling strategy efficiently and prove the applicability of this methodology to re-entry problems with large-scale PDEs.

The results indicate as well its superiority in terms of efficiency to a reduced discretization approach when compared with the application of OCPID-DAE1 to a nonlinear heat equation problem with a larger number of controls, and to the results provided in [47, 107] using a multiple shooting method. However, as discussed in Section 5.6, while our full discretization with structure exploitation strategy clearly offers a computational advantage in this case, it failed to solve optimal re-entry trajectory problems due to the ill-conditioning of the matrices that poses a hard challenge for the nonsmooth Newton method. The usual solution to this issue, a regularization of the matrix, did not improve the convergence either. In any case, we can conclude our analysis with an interesting insight: a reduced discretization approach can efficiently find optimal re-entry trajectories and solve coupled ODE-PDE problems when the number of controls is not large, and a full discretization approach is computationally superior when solving problems involving a large number of controls and PDEs of higher dimension, such as the active cooling problem considered here or potential TPS design problems.

These results are bounded by the models and problems considered here, and further research should explore the application of these methods to optimal control problems in different applications. In particular, more complex models of PDEs with applications to shape optimization and TPS design during re-entry would be of high interest in this context. The tailoring of the structure exploitation to the particular substructure yielded by the derivatives of the dynamics and constraints of other large-scale optimal control problems should also be considered. The iterative process of the forward-backward substitution presents a limitation to the efficiency of the substructure-exploiting methodology that could be improved by a parallel implementation.

Bibliography

[1] T. Akman, J. Diepolder, B. Grüter and F. Holzapfel. *Using Sensitivity Penalties to Robustify the Optimal Reentry Trajectory of a Hypersonic Vehicle.* In: ICAS 31st Congress of the International Council of the Aeronautical Science, Conference Proceedings, 2018.

[2] T. Akman, B. Hosseini, J. Diepolder, B. Grüter, R.J.M. Afonso, M. Gerdts and F. Holzapfel. *Efficient Sensitivity Calculation for Robust Optimal Control.* In: Deutscher Luft- und Raumfahrtkongress Conference Proceedings, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., 2020.

[3] A.Z. Al-Garni, A.Z. Sahin and B.S. Yilbas. *Active Cooling of a Hypersonic Plane Using Hydrogen, Methane, Oxygen and Fluorine.* Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 210 (1) (1996), pp. 9–17.

[4] A.Z. Al-Garni, A.Z. Sahin, B.S. Yilbas and S.A. Ahmed. *Cooling of aerospace plane using liquid hydrogen and methane.* Journal of Aircraft 32 (3) (1995), pp. 539–546.

[5] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney and D. Sorensen. *LAPACK Users' Guide.* Third Edition. Society for Industrial and Applied Mathematics: Philadelphia, PA, 1999.

[6] J.D. Anderson. *Fundamentals of Aerodynamics.* McGraw-Hill Education, 2010.

[7] J.D. Anderson. *Mathematische Optimierungsverfahren des Operations Research.* De-Gruyter: Berlin, 2010.

[8] J. Barlow and A.Z. Al-Garni. *The ascending trajectories performance and control to minimize the heat load for the transatmospheric aero-space planes.* In: 17th Atmospheric Flight Mechanics Conference, 1990.

[9] C. Bauer, A. Frink and R. Kreckel. *Introduction to the GiNaC Framework for Symbolic Computation within the C++ Programming Language*. Journal of Symbolic Computation 33 (1) (2002), pp. 1–12.

[10] M.S. Bazaraa, D.H. Sherali and C.M. Shetty. *Nonlinear programming: Theory and algorithms*. John Wiley and Sons: New York, 2006.

[11] J.T. Betts. *A collection of optimal control test problems*. 2015. URL: `http://appliedmathematicalanalysis.com`.

[12] J.T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, 2010.

[13] J.T. Betts and W.P. Huffman. *Exploiting Sparsity in the Direct Transcription Method for Optimal Control*. Computational Optimization and Applications 14 (2) (1999), pp. 179–201.

[14] H.G. Bock, E. Kostina and J.P. Schlöder. *On the Role of Natural Level Functions to Achieve Global Convergence for Damped Newton Methods*. In: System Modelling and Optimization, ed. by M.J.D. Powell and S. Scholtes. Springer US: Boston, MA, 2000, pp. 51–74.

[15] A. Brandis and C. Johnston. *Characterization of Stagnation-Point Heat Flux for Earth Entry*. In: 45th AIAA Plasmadynamics and Lasers Conference, 2014.

[16] A. Britzelmeier and M. Gerdts. *A Nonsmooth Newton Method for Linear Model-Predictive Control in Tracking Tasks for a Mobile Robot With Obstacle Avoidance*. IEEE Control Systems Letters 4 (4) (2020), pp. 886–891.

[17] M. Burger and M. Gerdts. *DAE Aspects in Vehicle Dynamics and Mobile Robotics*. In: Applications of Differential-Algebraic Equations: Examples and Benchmarks, ed. by S. Campbell, A. Ilchmann, V. Mehrmann and T. Reis. Springer International Publishing: Cham, 2019, pp. 37–80.

[18] C. Büskens and D. Wassel. *The ESA NLP Solver WORHP*. In: Modeling and Optimization in Space Engineering 73, ed. by G. Fasano and J.D. Pintér. Springer New York, 2013, pp. 85–110.

[19] C. Büskens and M. Gerdts. *Emergency Landing of a Hypersonic Flight System: A Corrector Iteration Method for admissible Real-Time Optimal Control Approximations.* In: Optimalsteuerungsprobleme in der Luft- und Raumfahrt, Workshop in Greifswald des Sonderforschungsbereichs 255: Transatmosphärische Flugsysteme, (München). 2003, pp. 51–60.

[20] E. Casas and K. Chrysafinos. *Analysis of the Velocity Tracking Control Problem for the 3D Evolutionary Navier–Stokes Equations.* SIAM Journal on Control and Optimization 54 (1) (2016), pp. 99–128.

[21] K. Chudej, H. J. Pesch, M. Wächter, G. Sachs and F. L. Bras. *Instationary heat constrained trajectory optimization of a hypersonic space vehicle by ODE-PDE-constrained optimal control.* In: Variational Analysis and Aerospace Engineering 33, Springer, Berlin, 2009, pp. 127–144.

[22] F.H. Clarke. *Optimization and Nonsmooth Analysis.* Society for Industrial and Applied Mathematics, 1990.

[23] A.R. Conn, N.I.M. Gould and P.L. Toint. *Trust Region Methods.* Society for Industrial and Applied Mathematics, 2000.

[24] A.N. Daryina and A.F. Izmailov. *Semismooth Newton method for quadratic programs with bound constraints.* Computational Mathematics and Mathematical Physics 49 (10) (2009), p. 1706.

[25] A. De Marchi and M. Gerdts. *Nonsmooth Newton's Method: Some Structure Exploitation.* In: Computational Science – ICCS 2019, Springer International Publishing, 2019, pp. 409–420.

[26] P. Deuflhard. *A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting.* Numerische Mathematik 22 (4) (1974), pp. 289–315.

[27] M. Dinkelmann. *Reduzierung der thermischen Belastung eines Hyperschallflugzeug durch optimale Bahnsteuerung.* Dissertation. Technische Universität München, Munich, 1997.

[28] M. Dinkelmann, M. Wächter and G. Sachs. *Modelling and simulation of unsteady heat transfer effects on trajectory optimization.* Mathematics and Computers in Simulation 53 (2000), pp. 389–394.

[29] D. Dirkx and E. Mooij. *Conceptual Shape Optimization of Entry Vehicles Applied to Capsules and Winged Fuselage Vehicles.* Springer Aerospace Technology, Springer International Publishing Switzerland, 2017.

[30] D. Dirkx and E. Mooij. *Continuous aerodynamic modelling of entry shapes.* In: AIAA Atmospheric Flight Mechanics conference, 2011.

[31] I. Duff and J. Reid. *The design of MA48: a code for the direct solution of sparse unsymmetric linear systems of equations.* ACM Trans. Math. Softw. 22 (1996), pp. 187–226.

[32] A. Fasano, D. Hömberg and L. Panizzi. *A mathematical model for case hardening of steel.* Mathematical Models and Methods in Applied Sciences 19 (2011).

[33] J. Fay and F. Riddell. *Theory of stagnation point heat transfer in dissociated air.* Journal of the Aeronautical Sciences 25 (2) (1958), pp. 73–85.

[34] A. Fischer. *A special Newton-type optimization method.* Optimization 24 (3-4) (1992), pp. 269–284.

[35] A. Fischer. *Solution of Monotone Complementarity Problems with Locally Lipschitzian Functions.* Math. Program. 76 (3) (1997), 513–532.

[36] A. Fischer and C. Kanzow. *On finite termination of an iterative method for linear complementarity problems.* Mathematical Programming 74 (3) (1996), pp. 279–292.

[37] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, 2003.

[38] R. Fletcher and S. Leyffer. *Nonlinear programming without a penalty function.* Mathematical Programming 91 (1999), pp. 239–269.

[39] R. Fletcher, S. Leyffer and P.L. Toint. *On the Global Convergence of a Filter–SQP Algorithm.* SIAM Journal on Optimization 13 (1) (2002), 44–59.

[40] M. Gerdts. *OCPID-DAE1 – Optimal Control and Parameter Identification with Differential-Algebraic Equations of Index 1 - User's Guide.* Engineering Mathematics, Department of Aerospace Engineering, Bundeswehr University Munich. 2013. URL: `http://www.optimal-control.de`.

[41] M. Gerdts. *Optimal Control of ODEs and DAEs.* De Gruyter: Berlin/Boston, 2012.

[42] M. Gerdts, G. Greif and H.J. Pesch. *Numerical optimal control of the wave equation: optimal boundary control of a string to rest in finite time.* Mathematics and Computers in Simulation 79 (2008), pp. 1020–1032.

[43] M. Gerdts, S. Horn and S.-J. Kimmerle. *Line search globalization of a semismooth Newton method for operator equations in Hilbert spaces with applications in optimal control.* Journal of Industrial and Management Optimization 13(1) (2017), pp. 47–62.

[44] M. Gerdts and M. Kunkel. *A nonsmooth Newton's method for discretized optimal control problems with state and control constraints.* Journal of Industrial and Management Optimization 4 (2008), pp. 247–270.

[45] M. Gerdts and I. Xausa. *Avoidance Trajectories Using Reachable Sets and Parametric Sensitivity Analysis.* In: System Modeling and Optimization, ed. by D. Hömberg and F. Tröltzsch. Springer Berlin Heidelberg, 2013, pp. 491–500.

[46] E.M. Gertz and S.J. Wright. *Object-Oriented Software for Quadratic Programming.* ACM Transactions on Mathematical Software 29 (2001), pp. 58–81.

[47] P.E. Gill, L.O. Jay, M.W. Leonard, L.R. Petzold and V. Sharma. *An SQP method for the optimal control of large-scale dynamical systems.* Journal of Computational and Applied Mathematics 120 (1) (2000), pp. 197–213.

[48] P.E. Gill and W. Murray. *Numerically stable methods for quadratic programming.* Mathematical Programming 14 (1) (1978), pp. 349–372.

[49] P.E. Gill, W. Murray and M.A. Saunders. *SNOPT: An SQP algorithm for large-scale constrained optimization.* SIAM Review 47 (1) (2005), pp. 99–131.

[50] P.E. Gill, W. Murray and M.A. Saunders. *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming.* 2018.

[51] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright. *Inertia-Controlling Methods for General Quadratic Programming.* SIAM Review 33 (1) (1991), pp. 1–36.

[52] P.E. Gill, W. Murray and M. Wright. *Practical optimization.* Academic Press: London, 1981.

[53] P.E. Gill and E. Wong. *Sequential Quadratic Programming Methods.* In: Mixed Integer Nonlinear Programming, ed. by J. Lee and S. Leyffer. Springer New York, 2012, pp. 147–224.

[54] D.E. Glass, A.D. Dilley and H.N. Kelly. *Numerical Analysis of Convection / Transpiration Cooling.* Journal of Spacecraft and Rockets 38 (1) (2001), pp. 15–20.

[55] D. Goldfarb and A. Idnani. *A numerically stable dual method for solving strictly convex quadratic programs.* Mathematical Programming 27 (1983), pp. 1–33.

[56] S.P. Han. *A globally convergent method for nonlinear programming.* Journal of Optimization Theory and Applications (22) (1977), pp. 297–309.

[57] M. Heinkenschloss. *Projected Sequential Quadratic Programming Methods.* SIAM Journal on Optimization 6 (2) (1996), pp. 373–417.

[58] R. Herzog and K. Kunisch. *Algorithms for PDE-Constrained Optimization.* GAMM-Mitteilungen 33 (2010), pp. 163 –176.

[59] R. Herzog, G. Stadler and G. Wachsmuth. *Directional Sparsity In Optimal Control Of Partial Differential Equations.* SIAM Journal on Control and Optimization 50 (2012), pp. 943–963.

[60] E.R. Hillje. *Entry Aerodynamics at Lunar Return Conditions Obtained from the Flight of Apollo 4 (AS-501).* Tech. rep. TN D-5399. NASA, 1969.

[61] E.R. Hillje. *Entry Flight Aerodynamics from Apollo Mission AS-202.* Tech. rep. TN D-4185. NASA, 1969.

[62] M. Hintermüller, K. Ito and K. Kunisch. *The Primal-Dual Active Set Strategy as a Semismooth Newton Method.* SIAM Journal on Optimization 13 (2002), pp. 865–888.

[63] M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich. *Optimization with PDE Constraints.* Springer Netherlands, 2009.

[64] M. Hinze and A. Rösch. *Discretization of Optimal Control Problems.* In: Constrained Optimization and Optimal Control for Partial Differential Equations, ed. by G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich and S. Ulbrich. Springer Basel, 2012, pp. 391–430.

[65] M. Holtmannspötter, A. Rösch and B. Vexler. *A priori error estimates for the space-time finite element discretization of an optimal control problem governed by a coupled linear PDE-ODE system.* Mathematical Control & Related Fields 11 (3) (2021), pp. 601–624.

[66] D. Hömberg, K. Krumbiegel and N. Togobytska. *Optimal control of multiphase steel production.* Journal of Mathematics in Industry 9 (1) (2019), p. 6.

[67] A. Huber. *Methoden zur Berechnung optimaler Rennlinien im dynamischen Grenz-bereich.* Dissertation. Neubiberg: Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, 2020.

[68] K. Ito and K. Kunisch. *Applications of Semi-smooth Newton Methods to Variational Inequalities.* In: Control of Coupled Partial Differential Equations, Birkhäuser Basel, 2007.

[69] K. Ito and K. Kunisch. *Semi–Smooth Newton Methods for Variational Inequalities of the First Kind.* ESAIM: Mathematical Modelling and Numerical Analysis 37 (1) (2010), pp. 41–62.

[70] A. Jameson and L. Martinelli. *Aerodynamic shape optimization techniques based on control theory.* In: Computational Mathematics Driven by Industrial Problems, Springer Berlin Heidelberg, 2000, pp. 151–221.

[71] C. Jänsch and A. Markl. *Trajectory optimization and guidance for a Hermes-type reentry vehicle.* In: AIAA Navigation and Control Conference, 1991.

[72] H. Jiang. *Global Convergence Analysis of the Generalized Newton and Gauss-Newton Methods of the Fischer-Burmeister Equation for the Complementarity Problem.* Mathematics of Operations Research 24 (1999), pp. 529–543.

[73] J.E. Johnson, R.P. Starkey and M.J. Lewis. *Aerothermodynamic Optimization of Reentry Heat Shield Shapes for a Crew Exploration Vehicle.* Journal of Spacecraft and Rockets 44 (4) (2007), pp. 849–859.

[74] W. Karush. *Minima of Functions of Several Variables with Inequality as Side Constraints.* Master thesis. Chicago, Illinois: University of Chicago, 1939.

[75] I. Kazufumi and K. Kunisch. *On a semi-smooth Newton method and its globalization.* Mathematical Programming 118 (2) (2009), pp. 347–370.

[76] S.-J. Kimmerle. *Necessary optimality conditions and a semi-smooth Newton approach for an optimal control problem of a coupled system of Saint-Venant equations and ordinary differential equations.* Pure and Applied Functional Analysis 1 (2) (2016), pp. 231–256.

[77] S.-J. Kimmerle. *Optimal Control of Mean Field Models for Phase Transitions.* IFAC Proceedings Volumes 45 (2) (2012). 7th Vienna International Conference on Mathematical Modelling, pp. 1107–1111.

[78] S.-J. Kimmerle and M. Gerdts. *Numerical optimal control of a coupled ODE-PDE model of a truck with a fluid basin.* In: Proceedings of the 10th AIMS International Conference, 2015, pp. 515–524.

[79] S.-J. Kimmerle, M. Gerdts and R. Herzog. *Optimal Control of an Elastic Crane-Trolley-Load System - A Case Study for Optimal Control of Coupled ODE-PDE Systems.* Mathematical and Computer Modelling of Dynamical Systems 24 (2018), pp. 182–206.

[80] W. Ko, L. Gong and R. Quinn. *Reentry Thermal Analysis of a Generic Crew Exploration Vehicle Structure.* Tech. rep. TM-2007-214607. NASA, 2007.

[81] H. Kreim, B. Kugelmann, H.J. Pesch and M.H. Breitner. *Minimizing the maximum heating of a re-entering space shuttle: An optimal control problem with multiple control constraints.* Optimal Control Applications and Methods 17 (1) (1996), pp. 45–69.

[82] C.K. Krishnaprakas. *Efficient Solution of Spacecraft Thermal Models Using Preconditioned Conjugate Gradient Methods.* Journal of Spacecraft and Rockets 35 (6) (1998), pp. 760–764.

[83] H. Kuczera, H. Hauck, P. Krammer and P. Sacher. *The German Hypersonics Technology Programme - Status 1993 and perspectives.* In: Proceedings of the 5th International Aerospace Planes and Hypersonics Technologies Conference, Munich, 1993.

[84] H.W. Kuhn and A.W. Tucker. *Nonlinear Programming.* In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press: Berkeley, California, 1951, pp. 481–492.

[85] M. Kunkel. *Nonsmooth Newton Methods and Convergence of Discretized Optimal Control Problems subject to DAEs.* Dissertation. Neubiberg: Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, 2012.

[86] I. Lasiecka and R. Triggiani. *Control theory for partial differential equations: continuous and approximation theories.* Cambridge University Press, 2000.

[87] A. De Marchi and M. Gerdts. *Free finite horizon LQR: A bilevel perspective and its application to model predictive control.* Automatica 100 (2019), pp. 299–311.

[88] C.D. Marley and J.F. Driscoll. *Modeling an Active and Passive Thermal Protection System for a Hypersonic Vehicle.* In: 55th AIAA Aerospace Sciences Meeting, Grapevine, Texas, 2017.

[89] B. Martens. *Necessary Conditions, Sufficient Conditions, and Convergence Analysis for Optimal Control Problems with Differential-Algebraic Equations.* Dissertation. Neubiberg: Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, 2019.

[90] M. Mayrhofer. *Notflugbahnen eines zweistufigen Hyperschall-Flugsystems ausgehend vom Trennmanöver.* Seminarbericht des Sonderforschungsbereichs 255: Transatmosphärische Flugsysteme (1996), pp. 109–118.

[91] M. Mayrhofer. *Verbesserung der Missionssicherheit eines zukünftigen zweistufigen Raumtransportsystems mittels Flugbahnoptimierung.* Dissertation. Technische Universität München, Munich, 2002.

[92] J. Michael, K. Chudej, M. Gerdts and J. Pannek. *Optimal Rendezvous Path Planning to an Uncontrolled Tumbling Target.* In: Automatic Control in Aerospace 19, 2013, pp. 347–352.

[93] A. Miele. *Theory of optimum aerodynamic shapes.* Academic Press Inc: New York, 1965.

[94] R.A. Minzner. *The 1976 standard atmosphere above 86-km altitude: Recommendations of task group 2 to COESA.* Tech. rep. SP-398. NASA, 1976.

[95] G. Naresh Kumar, M. Ikram, A.K. Sarkar and S.E. Talole. *Hypersonic flight vehicle trajectory optimization using pattern search algorithm.* Optimization and Engineering 19 (1) (2018), pp. 125–161.

[96] NASA. *Columbia Accident Investigation Board.* Tech. rep. 2003.

[97] NASA. *U.S. Standard Atmosphere.* Tech. rep. TM-X-74335. 1976.

[98] J. Nocedal and S. Wright. *Numerical optimization.* Series in operations research and financial engineering, Springer: New York, 2006.

[99] B. Pablos and M. Gerdts. *A reduced discretization approach for a re-entry optimal control problem with minimum heating.* In: International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions and Engineering, FAR 2019 Conference Proceedings, ESA Publications Division, 2019.

[100] B. Pablos and M. Gerdts. *Substructure exploitation of a nonsmooth Newton method for large-scale optimal control problems with full discretization.* Mathematics and Computers in Simulation (2021, Accepted).

[101] J.-S. Pang. *Newton's Method for B-differentiable Equations.* Mathematics of Operations Research 15 (2) (1990), pp. 311–341.

[102] J.E. Pavlosky and L.G. St. Leger. *Apollo Experience Report - Thermal Protection Subsystem.* Tech. rep. TN D-7564. NASA, 1974.

[103] J. Pearson and J. Gondzio. *Fast interior point solution of quadratic programming problems arising from PDE-constrained optimization.* Numerische Mathematik 137 (2017), pp. 1–41.

[104] F. Pescetelli, E. Minisci and R. Brown. *Re-entry trajectory optimization for a SSTO vehicle in the presence of atmospheric uncertainties.* In: 5th European Conference for Aeronautics and Space Sciences, EUCASS, Munich, 2013.

[105] H.J. Pesch, A. Rund, W. Wahl and S. Wendl. *On some new phenomena in state-constrained optimal control if ODEs as well as PDEs are involved.* Control and Cybernetics 39 (2010).

[106] G. Petit and B. Luzum. *IERS conventions (2010).* IERS Conventions Centre, Technical Note No. 36 (2010).

[107] L. Petzold, J.B. Rosen, P.E. Gill, L.O. Jay and K. Park. *Numerical Optimal Control of Parabolic PDEs Using DASOPT.* In: Large-Scale Optimization with Applications: Part II: Optimal Design and Control, Springer New York, 1997, pp. 271–299.

[108] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze and E.F. Mishchenko. *The Mathematical Theory of Optimal Processes.* John Wiley and Sons: New York, 1962.

[109] M.J. Powell. *A fast algorithm for nonlinearily constrained optimization calculation.* In: Lecture Notes in Mathematics 630, Numerical Analysis, Springer Berlin-Heidelberg-New York, 1978.

[110] L. Qi and J. Sun. *A nonsmooth version of Newton's method.* Mathematical Programming 58 (1993), pp. 353–367.

[111] G. Sachs and M. Dinkelmann. *Reduction of coolant fuel losses in hypersonic flight by optimal trajectory control.* Journal of Guidance, Control, and Dynamics 19 (6) (1996), pp. 1278–1284.

[112] M. Samà, K. Palagachev, A. D'Ariano, M. Gerdts and D. Pacciarelli. *Terminal Control Area Aircraft Scheduling and Trajectory Optimization Approaches.* ITM Web of Conferences 14 (3) (2017).

[113] J.A. Samareh. *A Multidisciplinary Tool for Systems Analysis of Planetary Entry, Descent, and Landing (SAPE).* Tech. rep. TM-2009-215950. NASA, 2009.

[114] J.M. Sanz-Serna and J.G. Verwer. *Convergence analysis of one-step schemes in the method of lines.* Applied Mathematics and Computation 31 (1989). Special Issue Numerical Ordinary Diferrential Equations (Proceedings of the 1986 ODE Conference), pp. 183 –196.

[115] J.M. Sanz-Serna, J.G. Verwer and W. Hundsdorfer. *Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations.* Numerische Mathematik 50 (1986), pp. 405–418.

[116] E.N. Sarmin and L.A. Chudov. *On the stability of the numerical integration of systems of ordinary differential equations arising in the use of the straight line method.* USSR Computational Mathematics and Mathematical Physics 3 (6) (1963), pp. 1537 –1543.

[117] K. Schittkowski. *The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function.* Numerische Mathematik 38 (1982), pp. 83–114.

[118] C. Specht, M. Gerdts and R. Lampariello. *Neighborhood Estimation in Sensitivity-Based Update Rules for Real-Time Optimal Control.* In: European Control Conference (ECC20), 2020.

[119] C. Specht, M. Gerdts and R. Lampariello. *Tube-Based Model Predictive Control for the Approach Maneuver of a Spacecraft to a Free-Tumbling Target Satellite.* In: Annual American Control Conference (ACC), 2018.

[120] J.L. Speyer and D.H. Jacobson. *Primer on Optimal Control Theory.* Society for Industrial and Applied Mathematics, 2010.

[121] M.D. Stuber, V. Kumar and P.I. Barton. *Nonsmooth exclusion test for finding all solutions of nonlinear equations.* BIT Numerical Mathematics 50 (4) (2010), pp. 885–917.

[122] D. Sun and L. Qi. *On NCP-Functions.* Computational Optimization and Applications 13 (1999), pp. 201–220.

[123] J. Sun and L. Zhang. *A globally convergent method based on Fischer-Burmeister operators for solving second-order cone constrained variational inequality problems.* Computers and Mathematics with Applications 58 (10) (2009), pp. 1936 –1946.

[124] K. Sutton and R.A. Graves. *A general stagnation-point convective-heating equation for arbitrary gas mixtures.* Tech. rep. TR R-376. NASA, 1971.

[125] F. Teschner, T. Akman, C. Mundt and F. Holzapfel. *Aerodynamic and Trajectory Studies on Optimal Earth Reentry of a Capsule.* In: International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions and Engineering, FAR 2019 Conference Proceedings, ESA Publications Division, 2019.

[126] F. Teschner and C. Mundt. *Numerische Untersuchungen der laminar-turbulenten Transition mit Hilfe der parabolisierten Stabilitätsgleichungen in Kombination mit einer Euler/Grenzschichtmethode zweiter Ordnung.* In: 18. STAB-Workshop, November 2017, Göttingen.

[127] F. Tröltzsch. *Optimal Control of Partial Differential Equations.* American Mathematical Society, 2010.

[128] M. Ulbrich. *Nonsmooth Newton-like Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces.* Habilitation thesis. Technical University Munich, 2003.

[129] M. Ulbrich, S. Ulbrich and L.N. Vicente. *A globally convergent primal-dual interior-point filter method for nonlinear programming.* Mathematical Programming 100 (2) (2004), pp. 379–410.

[130] R.J. Vanderbei. *Linear Programming: Foundations and Extensions.* International Series in Operations Research and Management Science, Springer, 2001.

[131] J.G. Verwer. *Convergence and order reduction of diagonally implicit Runge-Kutta schemes in the method of lines.* In: Numerical Analysis: Proceedings of the Dundee Conference on Numerical Analysis, 1986, pp. 220–237.

[132] J.G. Verwer and J.M. Sanz-Serna. *Convergence of method of lines approximations to partial differential equations.* Computing 33 (1984), pp. 395–313.

[133] N.X. Vinh, A. Busemann and R.D. Culp. *Hypersonic and Planetary Entry Flight Mechanics.* Ann Arbor: The University of Michigan Press, 1980.

[134] G. Vossen. *Switching Time Optimization for Bang-Bang and Singular Controls.* Journal of Optimization Theory and Applications 144 (2) (2009), pp. 409–429.

[135] A. Wächter and L.T. Biegler. *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.* Mathematical Programming 106 (1) (2006), pp. 25–57.

[136] M. Wächter. *Optimalflugbahnen im Hyperschallflug unter Berücksichtigung der instationären Aufheizung.* Dissertation. Technische Universität München, Munich, 2004.

[137] J.-H. Webert, P.E. Gill, S.-J. Kimmerle and M. Gerdts. *A study of structure-exploiting SQP algorithms for an optimal control problem with coupled hyperbolic and ordinary differential equation constraints.* Discrete and Continuous Dynamical Systems - S 11 (6) (2018), pp. 1259–1282.

[138] S. Wendl, H.J. Pesch and A. Rund. *On a state-constrained PDE optimal control problem arising from ODE-PDE optimal control.* In: Recent Advances in Optimization and its Applications in Engineering, 2010, pp. 429–438.

[139] R. Windhorst, M. Ardema, J. Bowles, R. Windhorst, M. Ardema and J. Bowles. *Minimum heating reentry trajectories for advanced hypersonic launch vehicles.* In: AIAA Guidance, Navigation, and Control Conference, 1997.

[140] M. Witzgall and K. Chudej. *Flight Path Optimization subject to Instationary Heat Constraints*. In: 7th Vienna International Conference on Mathematical Modelling (MATHMOD 2012), Wien: International Federation of Automatic Control. 2012, pp. 1141–1146.

[141] A. Zafarullah. *Application of the Method of Lines to Parabolic Partial Differential Equations With Error Estimates*. Journal of the ACM 17 (2) (1970), 294–302.