Modeling and optimal control of multiphysics problems using the finite element method

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von
Julian Andrej

Kiel 2019

Erstgutachter:	Prof. Dr.-Ing. Thomas Meurer
Zweitgutachter:	Prof. Dr. Lars Grüne
Prüfungsdatum:	15.03.19

Abstract

Interdisciplinary research like constrained optimization of partial differential equations (PDE) for trajectory planning or feedback algorithms is an important topic. Recent technologies in high performance computing and progressing research in modeling techniques have enabled the feasibility to investigate multiphysics systems in the context of optimization problems.

In this thesis a conductive heat transfer example is developed and techniques from PDE constrained optimization are used to solve trajectory planning problems. In addition, a laboratory experiment is designed to test the algorithms on a real world application.

Moreover, an extensive investigation on coupling techniques for equations arising in convective heat transfer is given to provide a basis for optimal control problems regarding heating ventilation and air conditioning systems.

Furthermore a novel approach using a flatness-based method for optimal control is derived. This concept allows input and state constraints in trajectory planning problems for partial differential equations combined with an efficient computation. The stated method is also extended to a Model Predictive Control closed-loop formulation. For illustration purposes, all stated problems include numerical examples.

Zusammenfassung

Fachübergreifende Forschung wie Optimierung von partiellen Differenzialgleichungen mit Schranken für Trajektorienplanungs- oder Regelungsalgorithmen ist ein wichtiges Thema. Aktuelle Technologien im Gebiet der Supercomputer und Modellierungstechniken erlauben die Analyse von Multiphysik Systemen im Bereich von Optimierungsproblemen.

In dieser Arbeit wird ein Beispiel anhand eines Wärmeleitprozesses entwickelt und Techniken der Optimierung von partiellen Differenzialgleichungen mit Schranken für Trajektorienplanungsprobleme angewandt. Zusätzlich wird ein Laborexperiment definiert um die beschriebenen Algorithmen an einer Applikation zu testen.

Desweiteren werden Kopplungstechniken für Multiphysiksysteme, speziell für die Kopplung von konvektivem Wr̈metransfer, untersucht und bewertet. Diese werden in weiteren Optimierungsproblemen, am Beispiel einer Klimatisierung eines Raumes, angewendet.

Darauf folgend wird ein neuer Ansatz zur Optimalsteuerung auf Basis einer flachheitsbasierten Methode vorgestellt. Dieses Konzept erlaubt es Eingangs- und Zustandsschranken in Trajektorienplanungsprobleme für partielle Differenzialgleichungen zu integrieren. Diese Methode wird mit dem Konzept der modellprediktiven Regelung erweitert und abschließend werden numerische Beispiele beschrieben und Ergebnisse diskutiert.

Acknowledgements

First, I want to thank my advisor Prof. Dr.-Ing. Thomas Meurer for his guidance throughout my research. He always supported my ideas and I could benefit from his knowledge when help was needed. I also want to acknowledge the funding of the Deutsche Forschungsgemeinschaft (DFG) by the grant ME 3231/3-1. Thanks to the DFG project collaborators from University of Bayreuth and University of Konstanz, I could also benefit from the broad interdisciplinary knowledge of Prof. Dr. Lars Grüne and Prof. Dr. Stefan Volkwein and their Ph.D. students Simon Pirkelmann and Luca Mechelli respectively.

Likewise I acknowledge the help of the `Firedrake` developers gathered around Dr. David Ham from Imperial College London. They always tried to help when assistance in regard to `Firedrake` was needed.

Furthermore I want to thank the whole group of the Chair of Automatic Control, especially my colleagues Andreas Kater, Philipp Rosenzweig and Dirk Siebelts. A lot of ideas evolved during our discussions in the office and it was always a fun and productive time working with them.

I want to thank my parents for always assisting me in any situation of life and most importantly my wife Marlena for supporting my decisions and her patience and encouragement during the process of this work.

Contents

List of Figures

List of Tables

Introduction

Numerical experiments are becoming more complex due to evolving technologies in high performance computing and progressing research in modeling, solving and preconditioning techniques [55]. Consequently the interest in solving problems that involve more than one physical phenomenon or the interaction between them is increasing [48]. Originally those multiphysics problems were handled by utilizing techniques like operator splitting to decouple the arising system and known solver and preconditioning techniques were used on the individual equations [27, 23]. The methods are also used in modern development, due to existing and well tested codes which require interaction with other simulations to enable multiphysics research [74]. Tightly coupled frameworks for simulation of all physics at once exist [31], but require reimplementation of the specific equations. A key to successful and accurate results in this area is the modeling approach of the coupling of the arising multiphysics system, since inappropriate methods can lead to physical incorrect outcome. Here one has to analyze the strength of coupling between the physics thoroughly to ensure sufficient accuracy. In optimization or optimal control approaches it is not common to analyze the coupling strength first and afterwards make reductions to the model. Therefore a decoupling approach for optimization of a coupled air flow and heat transport system to mitigate long solution times is presented in this work.

These theoretical investigations and software technologies enable interdisciplinary research like constrained optimization of partial differential equations for trajectory planning or feedback algorithms using Model Predictive Control (MPC). As an example for an industry relevant application a technique for cooling steel profiles is presented in [26] but no comparison with laboratory experiments is shown. Beginning with a single physics example an implementation of a parallel optimal control solver for trajectory planning is presented on a heating problem. Well known techniques from optimal control [36, 83] like adjoint based gradient computation are used and combined with modern solver techniques and a parallel computing approach. Physical applications that show working concepts in laboratory experiments and comparisons with the theoretical or simulated results are rare and therefore a validation on a laboratory experiment is presented.

In the context of multiphysics systems, an example which involves multiple physical phenomena, namely the fluid flow and temperature in a room as a result of heating ventilation and air conditioning (HVAC) is developed. The arising system consists of the incompressible Navier-Stokes equations coupled with the energy equation. Discretizing and solving just the coupled system for simulation is an active research topic and common techniques for efficient and stable computations are shown in this work. The equations describing the optimal control problem are very similar, or even equal, to those arising from the Boussinesq equations. These problems have been studied by the authors in [6, 43] using nondimensionalized formulations for semiconductor melts. In recent publications different techniques for optimal control on problems directly related to HVAC were considered and evaluated, e.g. a classic control theory approach, the Linear-quadratic regulator, was presented in [11, 16] where the stationary solution of the linearized Navier-Stokes equation was used. In this thesis, the decoupling of physics and their feasibility for the optimization problems is shown. Furthermore, the implementation uses similar optimal control techniques as the former single physics example and supports parallel computation as well and also considers the time-dependent and fully nonlinear incompressible flow computations.

Using the investigations of the coupling approaches an alternative method exploiting the concept of differential flatness [30] is studied. Originally based on the theory for finite dimensional systems, this technique has been extended to partial differential equations with examples in linear and nonliear diffusion-reaction systems [57, 54, 75], hyperbolic systems [64, 87] and flexible structures [76, 58]. An industry relevant example on trajectory planning, the heating of deep-drawing tools aiding manufacturing processes, has been presented in [10] and led to promising results. One generally uses a Gevrey function for the flat output and therefore addressing constraints in the input trajectory or in the state is generally not possible directly. To mitigate this drawback alternatives using splines [22, 79] and spectral methods using Legendre polynomials with Gauss-Legendre-Lobatto nodes [70] were developed. In this work, a novel concept on how to handle a flat output using an integrator chain description based on the concepts from [49] is extended to partial differential equations to solve optimization problems for trajectory planning. Several scenarios for a diffusion-reaction equation are shown as examples. Furthermore, the concept is extended and a closed-loop control for PDEs using MPC is derived.

In Chapter 1 an introduction to the finite element method is given to serve as a basis for further discussions in the next chapters.

Following Chapter 2 the heat equation is derived from the fundamental concepts of physics and the corresponding discretization in space is derived by using the finite element method in combination with suitable numerical time integra-

tion methods. This chapter also presents the details of the two implementations for solving the heat equation that have evolved in the course of this work. The implementations are validated using the presented validation and verification methods.

Subsequently the basics of fluid dynamics are summarized in Chapter 3 where the Navier-Stokes and Boussinesq equations are derived and a suitable and stable discretization for the Navier-Stokes equation is discussed.

In Section 3.4 the topic of multiphysics coupling is treated and different coupling techniques are presented for the Boussinesq and the general Navier-Stokes and heat transfer system. A numerical comparison of the coupling methods is given and the implementation is validated for the Navier-Stokes equation using the DFG benchmark of a flow around a cylinder as well as the Boussinesq equation with the MIT benchmark of a heated cavity by comparing key parameters of the results with published results.

In Chapter 4 an introduction on partial differential equation constrained optimization is given and followed by an example of a heating process of an aluminum block. The physical lab experiment served as a benchmark for a trajectory planning approach using the techniques of optimal control. A homogeneous and an inhomogeneous temperature profile scenario are presented as examples that show the complete process from numerical computation of the trajectory up to the measurements of the sensors on the lab experiments.

An alternative method using a novel approach, named flatness-based optimal control, is presented in Chapter 5. Here, the trajectory planning exploits the flatness property of the partial differential equation to derive an integrator chain system that reflects the flat output and its derivatives. A detailed explanation on the resulting algorithm is given and the results are backed up by numerical simulations.

Leveraging the concepts of Chapter 5 an extension to a Model Predictive Control approach is shown in Section 5.2. Using the concepts of the flatness-based optimal control algorithm, a method on how to apply the concept in combination with a model predictive control algorithm is derived. This allows formulations with stabilizing properties and the extension of the concepts in Chapter 5 to possible unstable systems.

Further remarks and a discussion of the gathered results along with an outlook on future research ideas is given in Chapter 6.

Chapter 1

Introduction to the finite element method

In the course of this work, several partial differential equations occur to describe different types of physical phenomena. In order to find a suitable numerical discretization in space a variety of methods exist. The Finite Difference Method [52, 82] can treat the equations directly in their strong differential form but lacks the flexibility for complex geometries involving curved boundaries. A suitable method is the Finite Volume Method [29] as it supports complex geometries and unstructured grids and seems a natural concept for fluid flow problems due to the conservation principles but can not be extended to higher-order accuracy on general unstructured grids (see e.g. [41]).

Due to the extensive mathematical background and flexibility the finite element method (see [13]) is used throughout. The method is based on a variational framework which is summarized in Section 1.1 to give a mathematical background from a functional analysis point of view. Section 1.2 describes the necessary concepts of the finite element method to understand how the numerical representations of matrices and vectors can be obtained from the weak formulation. To address the problems involving advection, Section 1.3 shows how to mitigate effects arising from a finite element formulation.

1.1 Variational framework

Let $\Omega \in \mathbb{R}^n$ denote a bounded domain in an n-dimensional space with sufficiently smooth boundary Γ. The closure $\bar{\Omega}$ of Ω is the union of the domain Ω with its boundary $\bar{\Omega} = \Omega \cup \Gamma$. As a basic example of a generic partial differential equation

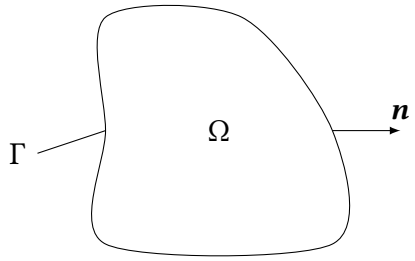


Figure 1.1: Domain Ω with boundary Γ and outward unit normal $\boldsymbol{n}$.

(PDE), consider the Poisson problem

$$-\nabla^2 u = f \qquad \text{in } \Omega, \tag{1.1}$$

$$u = u_\Gamma \qquad \text{on } \Gamma. \tag{1.2}$$

It follows a brief overview of the concepts from [33, 13] about the variational framework needed to describe PDE problems like Equation (1.1) using a weak formulation for the finite element method. A frequently needed function space is the Lebesgue space.

Definition 1.1 (Lebesgue space)**.** Let $\Omega \subset \mathbb{R}^n$ be a measurable set and let $f : \Omega \to \mathbb{R}$ be a measurable function. We define for a number $p \in (1, \infty]$ the **Lebesgue space**

$$L^p(\Omega) := \left\{ f : \|f\|_{L^p} < \infty \right\},$$

with the norm

$$\|f\|_{L^p} := \left(\int_\Omega |f(x)|^p \, dx \right)^{1/p}.$$

For $p = \infty$ the norm is denoted by

$$\|f\|_{L^\infty} := \operatorname*{ess\,sup}_\Omega |f(x)|_{L^p}.$$

With the multi-index notation $\alpha = (\alpha_1, \ldots, \alpha_N) \in \mathbb{N}^N$ and

$$|\alpha| = \sum_{i=1}^{N} \alpha_i,$$

denoting derivatives by

$$D^\alpha u = \frac{\partial^{\alpha_1} u}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_N} u}{\partial x_N^{\alpha_N}},$$

we describe functions that are not point wise differentiable in L^2 using the definition of the weak derivative.

Definition 1.2 (Weak derivative)**.** Denote the linear space of infinitely differentiable functions $C_0^\infty(\Omega)$ with compact support on Ω (see [33]) by

$$\mathcal{D}(\bar{\Omega}) = \{\phi \in C_0^\infty(\Omega)\}.$$

Consider $u \in L^2(\Omega)$ and $|\alpha| > 0$. If there exists $v \in L^2(\Omega)$ such that

$$\int_\Omega v(x)\phi(x)\,dx = (-1)^{|\alpha|}\int_\Omega u(x)D^\alpha\phi(x)\,dx \quad \forall\phi \in \mathcal{D}(\Omega),$$

then v is called the α-th **weak derivative** of u with $D^\alpha u := v$.

A frequently used function space along with Lebesgue space is a Hilbert space dependent on the following definition.

Definition 1.3 (Sobolev space)**.** For each integer $m \geq 0$ and real number p with $1 \leq p \leq \infty$, we define the **Sobolev space**

$$W^{m,p}(\Omega) = \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega)\ \forall|\alpha| \leq m\},$$

and the corresponding norms for $p \in (0, \infty]$

$$\|u\|_{W^{m,p}} = \left(\sum_{|\alpha|\leq m}\int_\Omega |D^\alpha u(x)|^p dx\right)^{1/p}$$

and for $p = \infty$

$$\|u\|_{W^{m,\infty}} = \max_{|\alpha|\leq m}\left(\operatorname*{ess\,sup}_{x\in\Omega}|D^\alpha u(x)|\right).$$

We denote by $H^m(\Omega) = W^{m,2}(\Omega)$, which is a Hilbert space equipped with the scalar product

$$(u, v)_m = \sum_{|\alpha|\leq m}\int_\Omega D^\alpha u(x)D^\alpha v(x)\,dx.$$

Let f a differentiable scalar function, then the gradient is defined as

$$\nabla f := \left[\frac{\partial f}{\partial x}, \ldots, \frac{\partial f}{\partial x_n}\right]^T.$$

Definition 1.4 (Greens identity)**.** Let Ψ, ϕ be at least twice differentiable functions then

$$\int_\Omega (\Psi\nabla^2\phi + \nabla\Psi\cdot\nabla\phi)\,dx = \int_\Gamma \Psi(\nabla\phi\cdot\boldsymbol{n})\,ds.$$

Assume V is a suitable function space (e.g. a Hilbert space) for the solution of u in Equation (1.1). Multiplying by a test function $v \in V$ and integrating by parts using Greens identity (c.f. Definition 1.4) yields

$$\int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx. \tag{1.3}$$

For notation purposes we introduce the definition of a bilinear form.

Definition 1.5 (Bilinear form)**.** A **bilinear form**, $a(\cdot, \cdot)$, on a linear space V is a mapping $a : V \times V \to \mathbb{R}$ such that each of the maps $v \mapsto a(v, w)$ and $w \mapsto a(v, w)$ is a linear form on V. It is symmetric if $a(v, w) = a(w, v)$ for all $v, w \in V$. An **inner product**, denoted by $(\cdot, \cdot)$, is a symmetric bilinear form on a linear space V that satisfies

$$(v, v) \geq 0 \, \forall \, v \in V \text{ and} \tag{a}$$
$$(v, v) = 0 \iff v = 0. \tag{b}$$

See [13] for the original definition.

Subsequently we can denote Equation (1.1) as a variational problem with the corresponding bilinear (c.f. Definition 1.5) and linear forms

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx, \tag{1.4}$$
$$(f, v) = \int_\Omega f v \, dx, \tag{1.5}$$

and can define a variational problem.

Problem 1.1. *Find $u \in V$ for a given $f \in V$ such that*

$$a(u, v) = (f, v) \quad \text{for all } v \in V. \tag{1.6}$$

Existence and uniqueness of the solution of Problem 1.1 can be determined by making use of the Lax-Milgram theorem [86]

Theorem 1.1 (Lax-Milgram)**.** *Let H be a Hilbert space, $a(\cdot, \cdot)$ a bilinear form with $a : H \times H \to \mathbb{R}$ and $L(v)$ a linear functional $L : H \to \mathbb{R}$. If and only if the following conditions*

(i) $a(\cdot, \cdot)$ is symmetric, $a(v, w) = a(w, v) \forall v, w \in H$,

(ii) $a(\cdot, \cdot)$ is elliptic, i.e., there exists $\alpha > 0$ s.t. $a(v, v) \geq \alpha \|v\|_H^2 \forall v \in H$,

(iii) $a(\cdot,\cdot)$ *is continuous, i.e., there exists a* $C \in \mathbb{R}^+$ *s.t.* $|a(v,w)| \leq C\|v\|_H\|w\|_H$,

(iv) L *is continuous, i.e., there exists a* $d \in \mathbb{R}^+$ *s.t.* $|L(v)| \leq d\|v\|_H$,

hold true, then there exists a unique solution to $a(u,v) = L(v)\forall v \in H$ *with the stability estimate* $\|u\|_H \leq d/C$.

With these prerequisites we can describe the Galerkin finite element method.

1.2 The finite element method

Using the weak formulation in Problem 1.1 in combination with a Galerkin projection method, the Hilbert space V needs to be split up, or discretized, into subspaces. This is achieved by defining a set of linearly independent functions which form a basis for V. The finite element method subdivides the continuous domain Ω into a finite set of finite elements (the definition originating from [18] for a finite element is used throughout this work) using a triangulation (see Definition 1.7).

Definition 1.6 (Finite element)**.** Let

(i) $K \subseteq \mathbb{R}^n$ be a bounded closed set with nonempty interior and piecewise smooth boundary (the **element domain**),

(ii) $\mathcal{P}$ be a finite-dimensional space of functions on K (the space of **basis** or **shape functions**) and

(iii) $\mathcal{N} = \{N_1, \ldots, N_k\}$ be a basis for the dual space $\mathcal{P}'$ (the set of **nodal variables**).

Then $(K, \mathcal{P}, \mathcal{N})$ is called a **finite element**.

Definition 1.7 (Triangulation)**.** A triangulation is described by the following properties.

(i) A subdivision of a domain Ω is a finite collection of open sets $\{\tau_i\}$ such that

  (i) $\tau_i \cap \tau_j = \emptyset \quad$ if $i \neq j$,

  (ii) $\bigcup \bar{\tau}_i = \bar{\Omega}$.

(ii) A subdivision $\mathcal{T} = \{\tau_i : 1 \leq i \leq m_{\mathcal{T}}\}$ with triangles τ_i, is a triangulation if no vertex of any triangle lies in the interior of some edges of another triangle. Furthermore, let $h_\tau := \sup_{x,y\in\mathcal{T}} \|x - y\|$ and ρ_τ the diameter of the largest possible circle inside of τ.

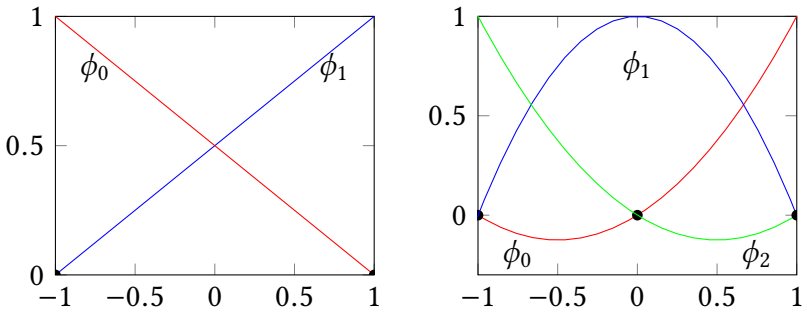


Figure 1.2: One dimensional Lagrange basis functions representing the linear (left) and quadratic (right) case.

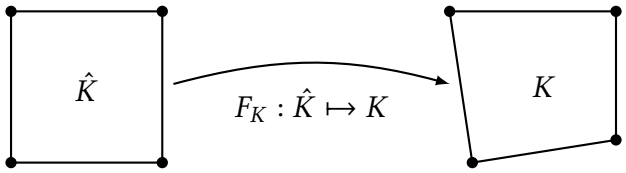


Figure 1.3: Illustration of the reference to physical map. Transformation of the coordinates in the reference frame $\hat{K}$ to the physical frame K.

(iii) The maximal mesh width h for a regular triangulation is defined by

$$h := \max_{\tau \in \mathcal{T}} h_\tau.$$

A simple example of a finite element is the Lagrange finite element defined on K (e.g. a quadrilateral). With Q_1 the set of linear Lagrange polynomials (see Figure 1.2 for a one dimensional example) in two dimensions we can define the finite element space

$$V_h = \{u_h \in H^1(\Omega) : u|_\tau \in Q_1 \text{ for all } \tau_i \in \mathcal{T}_h\}. \tag{1.7}$$

Transforming the integral over the domain Ω in Equation (1.3) to a sum of integrals over each cell in the triangulation $\mathcal{T}$ results in

$$\sum_{K \in \mathcal{T}} \int_K \nabla u \cdot \nabla v \, dx = \sum_{K \in \mathcal{T}} \int_K f v \, dx. \tag{1.8}$$

The finite element method relies on basis functions to approximate the solution on every element. These functions have to be computed based on the topology of the corresponding cell. It is common practice [88] to define these basis functions once on a so-called reference cell of the desired topology (e.g. a quadrilateral or triangle)

and use a one-to-one correspondence between the reference and the physical cell. Following the notation in [37] we define the reference cell $\hat{K}$. To describe the transformation of the reference cell to the physical cell in the triangulation $K \in \mathcal{T}$ we define a reference to physical map $F_K : \hat{K} \mapsto K$ (see Figure 1.3). The Pullback $H(\text{grad}, \hat{K}) \mapsto H(\text{grad}, K)$ is defined as the inverse transformation of the basis function on the reference cell $\hat{u} \circ F_K^{-1}$. Defining $J = \hat{\nabla} F_K$ the cell Jacobian matrix with $\hat{\nabla}$ the differentiation in reference coordinates. These representations allow the transformation of the cell wise integral in physical coordinates to reference coordinates

$$\int_\Omega \nabla u \cdot \nabla v \, dx = \int_{\hat{K}} J_K^{-T} \hat{\nabla} \hat{u} \cdot J_K^{-T} \hat{\nabla} \hat{v} |J_K| \, d\hat{x}. \tag{1.9}$$

With $\{\xi_q\}_{q=1}^{N_q}$ a set of quadrature points and quadrature weights $\{w_q\}_{q=1}^{N_q}$, the integral is approximated by a sum over the quadrature points and yields

$$\sum_q^{N_q} w_q \left(J_K^{-T}(\xi_q) \hat{\nabla} \hat{u}(\xi_q) \right) \cdot \left(J_K^{-T}(\xi_q) \hat{\nabla} \hat{v}(\xi_q) \right) |J_K(\xi_q)|. \tag{1.10}$$

Let us define $\hat{u}, \hat{v}$ on the polynomial basis in reference space $\hat{P}$ according to

$$\hat{u} = \sum_{i=1}^{N_f} u_i \hat{\phi}_i, \quad \hat{v} = \sum_{j=1}^{N_f} v_j \hat{\phi}_j, \tag{1.11}$$

and expanding the formulation to

$$\sum_{i,j=1}^{N_f} u_i v_j \underbrace{\sum_q^{N_q} w_q \left(J_K^{-T}(\xi_q) \hat{\nabla} \hat{\phi}_i(\xi_q) \right) \cdot \left(J_K^{-T}(\xi_q) \hat{\nabla} \hat{\phi}_j(\xi_q) \right) |J_K(\xi_q)|}_{A_e}. \tag{1.12}$$

The right hand side is described in a similar procedure with

$$\sum_{j=1}^{N_f} v_j \sum_q^{N_q} w_q f(F_K(\xi_q)) \hat{\phi}_j(\xi_q) |J_K(\xi_q)|. \tag{1.13}$$

For convenience, the approximated integral functions can be written in matrix vector notation

$$v^T A_e u = v^T f_e, \tag{1.14}$$

and in residual form

$$F(u) = v^T(A_e u - f_e) = 0, \tag{1.15}$$

with A_e the element stiffness (or Jacobian) matrix and $u = (u_0, \ldots, u_{N_f}), v = (v_0, \ldots, v_{N_f})$. Scaling out the test function v results in a linear system for the local element.

Remark 1. To obtain the stiffness matrix in a general formulation it is necessary to compute the Jacobian of the residual

$$\frac{\partial F(u)}{\partial u} = A_e. \tag{1.16}$$

Using the local element matrices A_e and vectors f_e combined with an appropriate mapping from the local to the global matrix vector structure, the global linear system is obtained (c.f. [88]) and can be solved using an appropriate numerical method.

1.3 Stabilization methods for advection dominated flows

In advection dominated equations, using the Galerkin finite element method described in Section 1.2 results in spurious oscillations in the solution (see e.g. [9]). To demonstrate the effect consider the time-dependent linear advection equation which models the transport of a scalar quantity q with constant velocity $\beta = 0.001$ on a one-dimensional domain $\Omega \in [0, 1]$

$$\frac{\partial q}{\partial t} + \beta \cdot \nabla q = 0 \qquad \text{in } \Omega, \tag{1.17a}$$

$$q = g \qquad \text{on } \Gamma, \tag{1.17b}$$

$$q = q_0 \qquad t = 0. \tag{1.17c}$$

Assume that the initial condition contains a discontinuous bump in the middle of the domain

$$q_0 = \begin{cases} 100 & 0.25 \leq x \leq 0.5, \\ 0 & \text{otherwise}, \end{cases}$$

with the boundary condition $g = 0$. Let us define the variational problem, which results from multiplying Equation (1.17) with a test function $v \in V$ and integrating over the domain. After applying the continuous Galerkin finite element method we arrive at

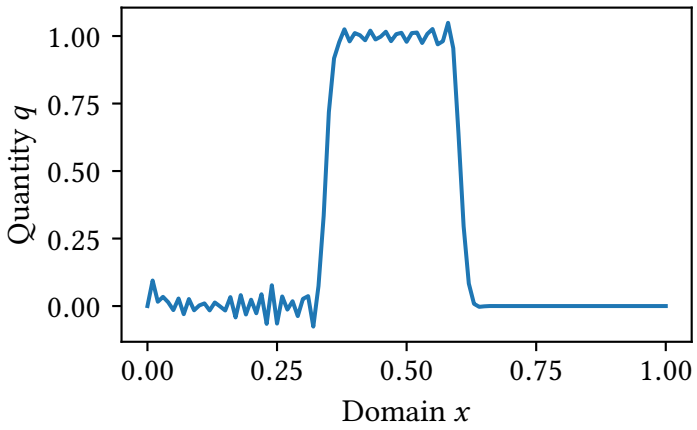


Figure 1.4: Advection dominated without stabilization at $t = 100.0$.

Problem 1.2. *Find $q_h \in V_h$ such that*

$$\left(\frac{\partial q_h}{\partial t}, v_h\right) + (\beta_h \cdot \nabla q_h, v_h) = 0 \quad \textit{for all } v_h \in V_h. \tag{1.18}$$

Solving this problem with a simple discretization in time (a forward Euler method in this case) we can see the solution in Figure 1.4 for a time step of $\Delta t = 1.0$. To prevent issues in the solution due to the time discretization we choose the parameters such that the Courant–Friedrichs–Lewy condition is less than 1.0 (see e.g. [9])

$$\text{CFL} = \frac{|\beta|\Delta t}{\Delta x} = \frac{0.001}{0.01} = 0.1.$$

It can be clearly seen in Figure 1.4 that spurious oscillations occur due to the spatial discretization method. To smooth the high frequency oscillations a stabilization using a diffusion in streamline direction, called Streamline-Upwind Petrov-Galerkin (SUPG) method, introduced in [15], is used. The stabilization term is composed of a residual of the corresponding equation and a non conforming test function

$$R = \frac{\partial q}{\partial t} - \beta \cdot \nabla q, \quad \bar{v} = \beta \cdot \nabla v.$$

Adding the terms to Problem 1.2 results in

Problem 1.3. *Find $q_h \in V_h$ such that*

$$(\frac{\partial q_h}{\partial t}, v_h) + (\beta_h \cdot \nabla q_h, v_h) + (R_h, \bar{v}_h) = 0 \quad \textit{for all } v_h \in V_h. \tag{1.19}$$

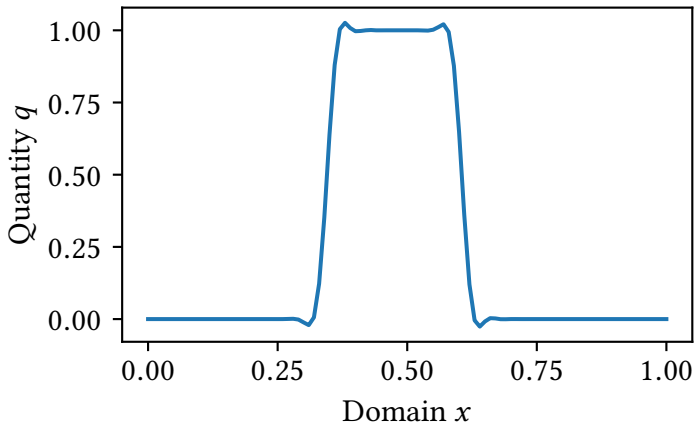


Figure 1.5: Advection dominated with SUPG stabilization at $t = 100.0$.

One can observe that if q_h is the solution the residual R vanishes which makes this method a consistent stabilization that does not harm the accuracy of the solution [15]. With the same parameters used in the non stabilized case, the solution of the stabilized case is shown in Figure 1.5, where the high frequency oscillations are damped. This simple example demonstrates the stabilization approach which is used in further chapters.

Remark 2. The author is aware of discontinuous Galerkin methods and corresponding high order techniques for advection dominated flow (see e.g. [41]). A recent detailed discussion and performance comparison of continuous and discontinuous Galerkin methods can be reviewed in [50]. Although these methods can be more robust due to their added Finite Volume Method characteristics through added internal element face contributions, they require more computational time. For scenarios expected from requirements of the code which is developed for this thesis, the discontinuous Galerkin discretization results in increased element stencil size in the discretization (see [46]) and would be slower. Slower means here that for the given order a discontinuous Galerkin discretization would take more time than a continuous Galerkin for the same accuracy achieved.

Chapter 2

Conductive heat transfer

The heat equation is a popular example when it comes to demonstration of algorithms regarding solution or optimal control of PDEs. Depending on the setting, this equation can lead to illustrative mathematical examples but can also address complex industry applications. Although desirable, physical applications that show working concepts in laboratory experiments and comparisons with the theoretical or simulated results are rare. As an example for an industry relevant example a technique for cooling steel profiles is presented in [26] but no comparison with laboratory experiments is shown. Recent results on trajectory planning for the heating of deep-drawing tools aiding manufacturing processes in the automobile industry are presented in [10]. The authors show a flatness-based trajectory planning approach and promising results when applied to the production environment. To address complex applications and show the process from stating a mathematical problem, we introduce the equations of conductive heat transfer as a prerequisite for further chapters.

In this chapter, the formulation of the heat equation is derived from general physical principles and suitable boundary conditions are presented. Section 2.3 shows the discretization of the heat equation using the finite element method and describes the two distinct implementations that evolved during this work. The implemented discretizations and solution methods are verified with the Method of Manufactured Solutions in Section 2.4.

2.1 Derivation of the heat equation

The partial differential equation describing heat distribution in a steady solid medium can be derived by using the first law of thermodynamics which states that the rate of change of total energy $\mathcal{E}$ stored in a control volume is equal to the rate of heat supplied $\dot{Q}$ to the volume minus the rate of work $\dot{W}$ done on the

system

$$\delta\mathcal{E} = \dot{Q} - \dot{W}.$$

Since we describe the solid as steady there is no kinetic energy E_{kin} in the total energy $\mathcal{E}$ and thus the internal energy E is

$$\delta\mathcal{E} = \delta E = \dot{Q} - \dot{W}.$$

Heat can be supplied through heat flux across the boundary due to conduction described by Fourier's Law that describes the heat flux with thermal conductivity k and temperature T as

$$\boldsymbol{q} = -k\nabla T.$$

Assuming a control volume of the steady material domain Ω with boundary Γ we describe the heat flux over the boundary in normal direction $\boldsymbol{n}$ as

$$\dot{W} = -\int_{\Gamma} \boldsymbol{q} \cdot \boldsymbol{n}\, ds,$$

where the negative sign describes that the energy decreases if heat flows out of the region. The heat generated in the volume is assumed to be known and described by Q. The rate of change of internal energy per unit mass can be described by the volume integral

$$\frac{\partial\mathcal{E}}{\partial t} = \int_{\Omega} \rho\frac{\partial E}{\partial t}\, dx$$

that describes the mass density of the specific material with ρ. Combining the terms and using the divergence theorem to transform the boundary integral to a volume integral results in

$$\rho\frac{\partial E}{\partial t} = \nabla \cdot k\nabla T + Q$$

With the simple equation of state, assuming that the temperature ranges are not too large, we can describe the linear relation between internal energy and temperature with

$$dE = C_p dT$$

and arrive at the partial differential equation of the heat equation

$$\rho C_p\frac{\partial T}{\partial t} + \nabla \cdot \boldsymbol{q} = Q, \tag{2.1}$$

where

- ρ is the density (SI $\mathrm{kg/m^3}$),
- C_p is the specific heat capacity at constant temperature (SI $\mathrm{J/(kg\,K)}$),
- T is the absolute temperature (SI K),
- k is the thermal conductivity (SI $\mathrm{m^2/s}$),
- $\boldsymbol{q}$ is the heat flux by conduction (SI $\mathrm{W/m^2}$),
- Q contains internal heat sources (SI $\mathrm{W/m^3}$).

2.2 Boundary conditions

The most typical boundary conditions for thermal applications are the specified temperature

$$T = T_s \quad \text{on } \Gamma_s \tag{2.2}$$

and the specified inward heat flux

$$-\boldsymbol{n} \cdot \boldsymbol{q} = Q_s \quad \text{on } \Gamma_s \tag{2.3}$$

where $\boldsymbol{n}$ is the outward normal vector of the boundary. If the boundary is insulated this results in

$$-\boldsymbol{n} \cdot \boldsymbol{q} = 0 \quad \text{on } \Gamma_s. \tag{2.4}$$

To describe the physics between the boundary and some temperature T_∞ far away from the domain the equation

$$-\boldsymbol{n} \cdot \boldsymbol{q} = h(T_\infty - T) \quad \text{on } \Gamma_s \tag{2.5}$$

with the heat transfer coefficient h with SI unit $\mathrm{W/(m^2\,K)}$ is used. The coefficient h approximates everything between the boundary and T_∞. This results from Newtons law of cooling. It simplifies a common case of radiation which is described by the Stefan-Boltzmann law

$$-\boldsymbol{n} \cdot \boldsymbol{q} = \epsilon\sigma(T_{amb}^4 - T^4) \quad \text{on } \Gamma_s \tag{2.6}$$

which results in nonlinearities.

2.3 Discretization

For the spatial discretization the finite element method described in Section 1.2 is used and the continuous finite element space V_h is defined for this problem as follows.

Impl.	Language	LA backend	Time disc.
MFEM	CPP	hypre	CVODE
Firedrake	python	PETSc	BDF-1

Table 2.1: Overview of the software implementations and used software frameworks.

Definition 2.1. Let Ω be a domain in $\mathbb{R}^n$ and $\mathcal{T}_h \subseteq \mathcal{T}$ a triangulation of Ω, then a continuous finite element space for the heat equation can be defined as

$$V_h = \left\{ v_h \in H^1 : q_h|_\tau \in \mathcal{P}_k \text{ for all } \tau_i \in \mathcal{T}_h \right\}.$$

A variety of finite element software frameworks have been evaluated. The implementation of Equation (2.1) is done in two frameworks. The first and mostly used here is built using a `python` implementation in combination with the `Firedrake` framework [66]. This approach enables quick testing by implementing the linear and bilinear forms directly in `UFL`, allows rapid prototyping of new methods and the deep integration with `PETSc` [5], enables easy testing of solver and preconditioner combinations. The second implementation uses a `CPP` framework called `MFEM` [62] which is coupled with the high performance parallel linear algebra backend of `hypre` [28]. This framework is also easily coupled with different ODE solvers to enable adaptive time stepping where appropriate. An overview of the two software combinations is shown in Table 2.1.

With the spatial discretization of Equation (2.1) and according boundary conditions we end up with a semi-discretized equation in space. For the discretization in time we use an adaptive time-stepping algorithm from the `SUNDIALS` suite [42] called `CVODE` which was first introduced in [19]. `CVODE` uses variable-order, variable-step multi step methods for the solution of initial value problems and encapsulates methods for stiff and non-stiff systems. For stiff systems the Backward Differentiation Formulas up to order 5 are used.

Since we use the finite element method for the spatial discretization, the derivation for the solution of the system in [42] has to be modified with a non-identity mass matrix. Consider an initial value problem with the variable $y \in \mathbb{R}^N$ and its time derivative $\dot{y}$, then the system of N ordinary differential equations we have to solve is

$$\dot{y} = M^{-1} f(t, y), \quad y(t_0) = y_0,$$

where M is the mass matrix arising from the finite element discretization. At every time step the (in general nonlinear) equation

$$G(y_n) = y_n - \gamma M^{-1} f(t_n, y_n) - a_n = 0$$

has to be solved. Here a_n and γ include coefficients depending on the time step, order of the method and history of step sizes. Stiff systems are solved by a series of Newton iterations (indexed with m and further explained in Algorithm 1) to obtain a solution for the current time step n. This requires the solution of a linear system

$$\left(I - \gamma M^{-1} J\right) \left[y_n^{m+1} - y_n^m\right] = -(y_n^m - \gamma M^{-1} f(t_n, y_n^m) - a_n) \tag{2.7}$$

with

$$\frac{\partial G}{\partial y} = I - \gamma M^{-1} J,$$
$$J = \frac{\partial f(t, y)}{\partial y}$$

at every time step. Modifying Equation (2.7) by multiplying with the mass matrix M from the left we obtain

$$(M - \gamma J) \left[y_n^{m+1} - y_n^m\right] = -M(y_n^m - \gamma M^{-1} f(t_n, y_n^m) - a_n), \tag{2.8}$$

which requires a mass matrix solve at every function evaluation of $\dot{y} = M^{-1} f(t, y)$ and a full time dependent Jacobian matrix solve. Therefore, no direct computation of the inverse mass matrix M^{-1} is necessary, which can be impossible in certain situations due to large system sizes.

2.4 Validation and verification

The validation and verification process is a very important step in the development of scientific software. Although, at a first glance, a solution generated by a non verified code may look correct, it is very common that later in the development process, when for example different parameter sets are evaluated, discrepancies in consistency of the solution may occur. There are several techniques to mitigate those issues and a comprehensive discussion on the topic can be found in [72]. Verification techniques look at whether the implemented equations are solved correctly and if the solutions are consistent for different parameter sets or boundary conditions. It is not concerned with the correctness of the physics implemented. One technique for verification is the method of manufactured solutions that prescribes a solution which is sophisticated enough and dependent on all parameters involved in the equation and may depend on time. This equation is used to compute a forcing function for the equation defined on the domain as well as the boundary conditions. It is possible to compute these with pen and paper but it is recommended to leave the work to a computer algebra system that derives a

symbolic solution and often is able to generate code for the corresponding input terms of the simulation. After the input functions are determined, a solution is generated from the code and the desired measures are computed. To test a spatial discretization a common procedure is to compute an error norm, e.g. the l_2, L_2 or H^1 norm of the obtained solution T and the manufactured solution T^* for increasing spatial accuracies. A positive test would show a convergence rate of the error norm according to mathematical analysis dependent on the function space and finite element combinations used. This test should be repeated with different sets of parameters to ensure the consistency of the solutions.

Consider the semi-discretized scenario of a heating problem that includes a specified inward heat flux on Γ_i and a cooling boundary condition on Γ_c which has to be modified to include a source term such that

$$\rho C_p \frac{\partial T}{\partial t} - k\nabla^2 T = Q \qquad \text{in } \Omega \times \mathbb{R}, \tag{2.9a}$$

$$\boldsymbol{n} \cdot k\nabla T = Q_i \qquad \text{on } \Gamma_i \times \mathbb{R}, \tag{2.9b}$$

$$\boldsymbol{n} \cdot k\nabla T = h(T_\infty - T) + Q_c \qquad \text{on } \Gamma_c \times \mathbb{R}. \tag{2.9c}$$

This scenario includes every boundary condition type that is implemented in the code and thus should cover all aspects to be tested. A suitable and sophisticated manufactured solution on the domain $\Omega \in [0,1] \times [0,1]$ with boundary $\Gamma = \Gamma_i \cup \Gamma_c$ for Equation (2.1) is for example

$$T^*(t) = \frac{\rho}{C_p} T_0 (1 + \sin(x)^2 \sin(2y)^2) \exp\left(\frac{t(t_0 - t)}{t_0}\right), \tag{2.10}$$

where T_0 can include different parameter combinations or an arbitrary constant. This equation is put into Equation (2.9) and the corresponding forcing terms Q, Q_i and Q_c are computed with a computer algebra system. To compute results the `MFEM` framework is used and the weak form of Equation (2.9) is then discretized with linear finite elements in space and the `CVODE` time-integration library is used for the resulting initial value problem, this yields:

Problem 2.1. *Find $T_h \in V_h$ with $t \in (0, t_e]$ where $t_e = 0.5$ such that for all $v_h \in V_h$*

$$\int \rho C_p \frac{\partial T_h}{\partial t} v_h \, d\Omega - \int k\nabla T_h \cdot \nabla v_h \, d\Omega = \int Q v_h \, d\Omega \qquad \text{in } \Omega \times \mathbb{R},$$

$$\int \boldsymbol{n} \cdot k\nabla T v_h \, d\Gamma_i = \int Q_i v_h \, d\Gamma_i \qquad \text{on } \Gamma_i \times \mathbb{R},$$

$$\int \boldsymbol{n} \cdot k\nabla T v_h \, d\Gamma_c - \int h(T_\infty - T) v_h \, d\Gamma_c = \int Q_c v_h \, d\Gamma_c \qquad \text{on } \Gamma_c \times \mathbb{R}.$$

El. order	Ref. level	L_2-error	Conv. rate
1	1	9.572×10^{-3}	2.0
1	2	2.415×10^{-3}	2.0
1	3	6.081×10^{-4}	2.0
1	4	1.523×10^{-4}	2.0
2	1	5.769×10^{-3}	3.0
2	2	7.256×10^{-4}	3.0
2	3	9.083×10^{-6}	3.0
2	4	1.134×10^{-6}	3.0

Table 2.2: L_2-error and convergence rates for the manufactured solution test problem using `MFEM` framework.

El. order	Ref. level	L_2-error	Conv. rate
1	1	1.260×10^{-2}	2.0
1	2	3.194×10^{-3}	2.0
1	3	8.015×10^{-4}	2.0
1	4	2.006×10^{-4}	2.0
2	1	6.793×10^{-4}	3.0
2	2	8.660×10^{-5}	3.0
2	3	1.088×10^{-5}	3.0
2	4	1.362×10^{-6}	3.0

Table 2.3: L_2-error and convergence rates for the manufactured solution test problem using `Firedrake` framework.

The absolute tolerance of `CVODE` is chosen as 10^{-8} and does not interfere with the spatial convergence of the analysis and the algorithm. The `CVODE` implementation is assumed to be tested and verified. Computed L_2-error and convergence rates for the manufactured solution test problem can be found in Table 2.2. The same tests were done with the implementation using `Firedrake` combined with the BDF-1 time stepping algorithm and the results are denoted in Table 2.3. Both results agree with the theoretical convergence rate of the finite element order $p + 1$ and verify the correctness of the code with the used boundary conditions.

Chapter 3

Convective heat transfer

In the following the derivation of a suitable model for simulating air flow and temperature transport in a room, for example an office space, is considered. A suitable model has to cover air flows from windows, open doors or a ventilation system. Associated with these is the exchange of thermal energy heating or cooling the room. Therefore the Navier-Stokes equations are derived to serve as a foundation for air flow. Consequently these equations are coupled with heat transport. Discretization of the Navier-Stokes equation is not a trivial task and involves thorough analysis. A common technique covering that problem is reviewed in Section 3.3. As described in Section 3.4 it is possible to formulate different versions of the coupled problem depending on the temperature setting. Moreover a guide on the numerical solution is given in Section 3.4.2 as well as a numerical comparison on different coupling methods in Section 3.4.3.

The implementation of the solution method of the Navier-Stokes equations is tested on the DFG benchmark of a flow around a cylinder in Section 3.5.1. The more sophisticated solution of the fully-coupled Boussinesq equation, which covers fluid flow in combination with heat transport, is validated in Section 3.5.2.

3.1 Derivation of the Navier-Stokes equations

The fluid is treated as a continuum to derive equations describing the flow velocity $\boldsymbol{u}$, thus molecular structure and motion are ignored. The references [7, 3, 23] describe how to derive the governing differential equations by looking at the three concepts of:

- conservation of mass,
- Newtons second law of motion,
- the first law of thermodynamics (conservation of energy).

An important concept to derive the conservation laws is the notion of the material derivative described in [7] that expresses a differentiation following the motion of the fluid as an operator

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \nabla. \tag{3.1}$$

Consider the fluid domain Ω. The mass enclosed by the boundary Γ of Ω is at every point in time the integral of the density per unit mass over the volume and the outflow quantities are described by the rate of mass flowing outwards across the surface. In mathematical notation this results in absence of any mass source in

$$\int_\Omega \rho \, dx + \int_\Gamma \rho \boldsymbol{u} \cdot \boldsymbol{n} \, ds = 0,$$

and can be transformed to a volume integral

$$\int_\Omega \frac{\partial \rho}{\partial t} \, dx + \int_\Omega \nabla \cdot (\rho \boldsymbol{u}) \, dx = 0,$$

which has to hold for every chosen volume. If the density ρ does not depend on the pressure and thus does not change in time the latter expression reduces to the so-called incompressibility condition

$$\nabla \cdot \boldsymbol{u} = 0.$$

To describe the motion of a fluid we denote the rate of change of the momentum $\rho \boldsymbol{u}$ with the material derivative integrated over a part of the fluid volume

$$\int_\Omega \left(\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) dx.$$

The total volume forces are denoted by $\boldsymbol{f}$ and the stress tensor $\boldsymbol{T}$, which represent every surface force acting on the control volume of the fluid. Therefore, we can write the momentum balance in the form

$$\int_\Omega \left(\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) dx = \int_\Omega \rho \boldsymbol{f} \, dx + \int_\Omega \nabla \cdot \boldsymbol{T} \, dx,$$

which has to hold for every choice of fluid volume Ω and thus requires

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} = \rho \boldsymbol{f} + \nabla \cdot \boldsymbol{T}.$$

The stress tensor $\boldsymbol{T}$ describes the mechanical definition of pressure as a volumetric stress and deviatoric stress with

$$\boldsymbol{T} = (-p + \lambda \operatorname{tr}(\boldsymbol{D}))\mathbb{I} + 2\mu \boldsymbol{D},$$

with the symmetric strain rate tensor

$$\boldsymbol{D} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right).$$

The constant λ can be determined by taking into account the Stokes hypothesis, which states that the bulk viscosity vanishes, holds and thus

$$2\mu + 3\lambda = 0, \quad \lambda = \frac{2}{3}\mu,$$

with the kinematic viscosity μ of the fluid. The last needed quantity is the internal energy E which is derived using the first law of thermodynamics stating that the rate of change of total energy $\mathcal{E}$ of a fluid part is equal to the rate of heat supplied to the system minus the rate of work done on the system,

$$\delta \mathcal{E} = \dot{Q} - \dot{W}.$$

To derive the internal energy we note that the total energy consists of kinetic energy E_{kin} and internal energy E [7, 63]

$$\delta \mathcal{E} = \delta E_{kin} + \delta E.$$

The whole work done on the system is defined by the change of kinetic energy and the influence of all body and surface forces acting on the material. Neglecting heat sources inside the material, the only way heat can be supplied is through heat flux across the boundary due to conduction described by Fourier's Law that describes the heat flux as

$$\boldsymbol{q} = -k \nabla T.$$

Then the total rate of work done on the fluid volume is (c.f. [7])

$$W = \frac{1}{\rho} \boldsymbol{T} : \boldsymbol{D},$$

and the rate of heat supplied is

$$Q = -\frac{1}{\rho} \nabla \cdot \boldsymbol{q}.$$

Furthermore we describe the internal energy balance with

$$\rho \frac{DE}{Dt} = \boldsymbol{T} : \boldsymbol{D} + \nabla \cdot k \nabla T.$$

Expanding the dyadic expressions results in

$$\rho \frac{DE}{Dt} = -p \nabla \cdot \boldsymbol{u} + \nabla \cdot k \nabla T + \boldsymbol{\Phi},$$

where

$$\boldsymbol{\Phi} = \left(\frac{2}{3} \mu + \lambda \right) \operatorname{tr} (\boldsymbol{D})^2 + 2 \mu \boldsymbol{D}' : \boldsymbol{D}'$$

is the dissipation of energy due to viscous effects. We assume the ideal gas law and can describe the temperature and pressure using the equations of state

$$\begin{aligned} dE &= C_v T, \\ p &= (\gamma - 1) \rho E \end{aligned}$$

with the specific heat constants C_v, C_p and ratio $\gamma = \frac{C_v}{C_p}$. From these relations, one can observe that in an ideal gas the internal energy is only dependent on the temperature. Using these prerequisites we describe the compressible Navier-Stokes equations

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} = \rho \boldsymbol{f} + \nabla \cdot \boldsymbol{T} \qquad \text{in } \Omega \times \mathbb{R}, \tag{3.2a}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \text{in } \Omega \times \mathbb{R}, \tag{3.2b}$$

$$\rho \frac{\partial E}{\partial t} + \rho \boldsymbol{u} \cdot \nabla E = -p \nabla \cdot \boldsymbol{u} + \nabla \cdot k \nabla T + \boldsymbol{\Phi} \qquad \text{in } \Omega \times \mathbb{R}. \tag{3.2c}$$

If the incompressibility condition holds, Equations (3.2a) to (3.2c) reduce to the incompressible Navier-Stokes equations

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho (\boldsymbol{u} \cdot \nabla) \boldsymbol{u} = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \rho \boldsymbol{f} \qquad \text{in } \Omega \times \mathbb{R}, \tag{3.3a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \times \mathbb{R}, \tag{3.3b}$$

which are independent of the energy equation. The SI units of the used quantities are described below:

- ρ is the density (SI kg/m^3),
- $\boldsymbol{u}$ is the vector valued velocity (SI m/s),
- μ is the kinematic viscosity (SI kg/ms),
- p is the pressure (SI Pa),
- $\boldsymbol{f}$ contains internal forces on the fluid (SI m/s).

Looking at Equations (3.3a) and (3.3b), there is no relation between temperature changes and the resulting volume force in the velocity of the fluid due to the

assumption of a constant density. Some applications still show effects of natural convection due to the gravity acceleration $\boldsymbol{g}$ and are assumed incompressible. In relevant applications, for example heating or cooling of a room in a home or office environment where airflow velocities are in the range for $3\,\mathrm{m\,s^{-1}}$ to $5\,\mathrm{m\,s^{-1}}$ the velocity of the fluid is in range of $\boldsymbol{g}$. This effect is approximated by a Buoyancy force which can be modeled with a volume force term in Equation (3.3a) only dependent on the gravity acceleration $\boldsymbol{g}$ the temperature difference relative to a reference temperature T_0 and a thermal expansion coefficient α [$\mathrm{K^{-1}}$]. Thus the change in density is approximated by

$$\Delta\rho\boldsymbol{g} \approx -\rho\boldsymbol{g}\alpha(T - T_0). \tag{3.4}$$

Using Equation (3.4) as a volume force in Equation (3.3a) results in the Boussinesq equations

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\nabla p + \mu\nabla^2\boldsymbol{u} - \rho\boldsymbol{g}\alpha(T - T_0) \qquad \text{in } \Omega\times\mathbb{R}, \tag{3.5a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega\times\mathbb{R}, \tag{3.5b}$$

$$\rho C_p\frac{\partial T}{\partial t} + \rho C_p\boldsymbol{u}\cdot\nabla T = \nabla\cdot k\nabla T \qquad \text{in } \Omega\times\mathbb{R}. \tag{3.5c}$$

According to [29] this approximation is valid especially for $\alpha(T - T_0) < 1$ and for temperature differences of $\Delta T = 2\,\mathrm{K}$ for water and $\Delta T = 15\,\mathrm{K}$ for air.

3.2 Boundary conditions

When analyzing flow problems most scenarios include a typical set of boundary conditions which utilize at least an inflow, outflow and a wall. In this work, we restrict the presentation to these boundary conditions but more complicated conditions can be formulated.

A fluid near a wall is assumed to have the same velocity as the wall itself. If the wall does not move a no-slip boundary condition is introduced which sets the velocity via a Dirichlet type at the wall to zero using

$$\boldsymbol{u} = 0 \quad \text{ on } \Gamma_w.$$

An inflow with known velocity is mostly introduced by prescribing the velocity directly

$$\boldsymbol{u} = \boldsymbol{u}_{in} \quad \text{ on } \Gamma_{in}.$$

When describing an outlet there are more possibilities. Most commonly one does not know the velocity at the outlet interface and thus a direct assignment like in

the inflow is not possible. An easy way to describe the outflow is to prescribe the pressure at the interface with

$$p = 0 \quad \text{on } \Gamma_{out},$$

which basically introduces a no stress condition at the interface for the fluid. This method is not always convenient for the numerical method which are discussed in Section 3.3. Therefore the stress-free (also called do nothing) boundary condition for outflows is introduced by imposing no traction on the boundary with

$$\boldsymbol{T} \cdot \boldsymbol{n} = 0 \qquad \text{on } \Gamma_{out},$$

or equivalently

$$-p\boldsymbol{n} + \lambda \operatorname{tr}(\boldsymbol{D})\boldsymbol{n} + 2\mu \boldsymbol{D} \cdot \boldsymbol{n} = 0 \qquad \text{on } \Gamma_{out},$$

which yields for the incompressible form

$$\mu \nabla \boldsymbol{u} \cdot \boldsymbol{n} - p\boldsymbol{n} = 0 \quad \text{on } \Gamma_{out}.$$

3.3 Discretization

Finding a stable and performant discretization of the Navier-Stokes equations can be a difficult topic. Depending on the scenarios and the resulting flow fields one wants to analyze, a proper choice has to be made. Herein the finite element method is used for the spatial discretization. Since all scenarios in this work include only incompressible flow we use Equations (3.3a) and (3.3b) to derive the variational form by multiplying with a test function and integrating by parts. Looking at the terms separately we have to introduce two operators that are differentiable with $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ and $n = 2, 3$ the divergence of a vector field and dyadic product of two tensors are denoted by

$$\nabla \cdot \boldsymbol{u} := \frac{\partial u_i}{\partial x_i} + \cdots + \frac{\partial u_n}{\partial x_n},$$
$$\nabla \boldsymbol{u} : \nabla \boldsymbol{v} := \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j}.$$

For illustration purposes, assume only Dirichlet boundary conditions on Γ. Let us introduce the corresponding function spaces for the test function $\boldsymbol{v}$ of the velocity $\boldsymbol{u}$ and the test function q of the pressure p as

$$\boldsymbol{V} = \{\boldsymbol{v} \in \boldsymbol{H}^1,\ \boldsymbol{v}|_\Gamma = 0\} \quad \text{and} \quad Q = \{q \in L_2\}.$$

Multiplying Equation (3.3a) by $\boldsymbol{v}$ and Equation (3.3b) by q, integrating over the domain, yields after reordering the terms

$$\int_\Omega \left[\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \mu\nabla^2\boldsymbol{u} - \rho\boldsymbol{f}\right]\cdot\boldsymbol{v}\,dx = 0, \tag{3.6a}$$

$$\int_\Omega \nabla\cdot\boldsymbol{u}q\,dx = 0. \tag{3.6b}$$

Looking at specific terms of the first equation separately and define the following bilinear and linear forms using the inner product notation (see Definition 1.5). The bilinear form

$$a(\boldsymbol{u},\boldsymbol{v}) := \int_\Omega \mu\nabla\boldsymbol{u} : \nabla\boldsymbol{v}\,dx \tag{3.7}$$

results from integrating by parts and using Greens identity (see Definition 1.4). The remaining forms are

$$b(\boldsymbol{v},q) := \int_\Omega q\nabla\cdot\boldsymbol{v}\,dx, \tag{3.8}$$

$$c(\boldsymbol{u},\boldsymbol{v}) := \int_\Omega \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u})\cdot\boldsymbol{v}\,dx. \tag{3.9}$$

With these prerequisites we can now formulate a variational problem consisting of Equations (3.3a) and (3.3b) that reads:

Problem 3.1. *Find $\boldsymbol{u}\in\boldsymbol{V}$ and $p\in Q$ such that*

$$\left(\rho\frac{\partial\boldsymbol{u}}{\partial t},\boldsymbol{v}\right) + c(\boldsymbol{u},\boldsymbol{v}) + a(\boldsymbol{u},\boldsymbol{v}) - b(\boldsymbol{v},p) - (\rho\boldsymbol{f},\boldsymbol{v}) = 0 \qquad \text{for all } \boldsymbol{v}\in\boldsymbol{V}, \tag{3.10a}$$

$$b(\boldsymbol{u},q) = 0 \qquad \text{for all } q\in Q. \tag{3.10b}$$

The next step is to find a suitable finite element pair that discretizes the mixed function space of $\boldsymbol{V}\times Q$. Looking at the literature (especially [13, 14]), these pairs have to fulfill certain conditions in order to give a meaningful solution. A fundamental result is the following lemma,

Lemma 3.1 (Inf-sup condition)**.** *In order for a finite element discretization of Equations* (3.3a) *and* (3.3b) *to have a unique solution, it is necessary and sufficient that*

$$0 < \beta := \inf_{\boldsymbol{v}\in\boldsymbol{V}_h, q\in Q_h}\sup\frac{|b(\boldsymbol{v},q)|}{\|\boldsymbol{v}\|_V\|q\|_Q}. \tag{3.11}$$

Proof. See [13, p. 342]. □

Figure 3.1: $(\mathcal{P}_2/\mathcal{P}_1)$ Taylor-Hood finite elements in $\mathbb{R}^n$.

From this we conclude, that if a finite element pair $(\boldsymbol{V}_h, Q_h)$ fulfills the inf-sup condition, then a solution exists and is unique (see [13, 34]). An example of a stable finite element pair is presented in the following definition and is depicted in Figure 3.1.

Definition 3.1 (Taylor-Hood)**.** Let Ω be a domain in $\mathbb{R}^n$ and $\mathcal{T}_h \subseteq \mathcal{T}$ be a triangulation of Ω in a finite number of simplexes. Then the **Taylor-Hood** finite element space is denoted by

$$\boldsymbol{V}_h = \{\boldsymbol{v}_h \in \boldsymbol{H}^1 : \boldsymbol{v}|_\tau \in (\mathcal{P}_{k+1})^N \text{ for all } \tau_i \in \mathcal{T}_h\}, \tag{3.12a}$$

$$Q_h = \left\{q_h \in L_2 : q_h|_\tau \in \mathcal{P}_k \text{ for all } \tau_i \in \mathcal{T}_h, \int_\Omega q_h \, dx = 0\right\}. \tag{3.12b}$$

Definition 3.1 was introduced in [80] including an extensive stability analysis. Other stable finite elements exist, but are not considered in this case. As a result, we define the discretized variational form of Problem 3.1 using Equation (3.12). This yields

Problem 3.2. *Find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ and $p_h \in Q_h$ such that*

$$\left(\rho \frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}_h\right) + c(\boldsymbol{u}_h, \boldsymbol{v}_h) + a(\boldsymbol{u}_h, \boldsymbol{v}_h) - b(\boldsymbol{v}_h, p_h) - (\rho \boldsymbol{f}_h, \boldsymbol{v}_h) = 0, \tag{3.13}$$

$$b(\boldsymbol{u}_h, q_h) = 0, \tag{3.14}$$

for all $\boldsymbol{v}_h \in \boldsymbol{V}_h$ and $q_h \in Q_h$.

As mentioned in Section 1.3 the flow in Problem 3.2 can be advection dominated and in such cases even a stable finite element pair leads to high oscillations in the solution. Therefore we adapt the Streamline-Upwind Petrov-Galerkin (SUPG) method for the Navier-Stokes equation from [81]. For notation purposes let us define the residual formulation of Problem 3.2 with

$$\mathcal{A}(\boldsymbol{u}_h, \boldsymbol{v}_h, p_h, q_h) = (3.13) - (3.14) = 0, \tag{3.15}$$

and the residual formulation of the incompressible Navier-Stokes equation with

$$\mathcal{R}(\boldsymbol{u}_h, p_h) = (3.3a) - (3.3b) = 0. \tag{3.16}$$

Now with the SUPG stabilization term

$$\tau_{SUPG}(\boldsymbol{u}_h \cdot \nabla)\boldsymbol{v}_h \tag{3.17}$$

we define the stabilized problem:

Problem 3.3. *Find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ and $p_h \in Q_h$ such that*

$$\mathcal{A}(\boldsymbol{u}_h, \boldsymbol{v}_h, p_h, q_h) + \sum_{K \in \mathcal{T}} \int \tau_{SUPG}(\boldsymbol{u}_h \cdot \nabla)\boldsymbol{v}_h \cdot \mathcal{R}(\boldsymbol{u}_h, p_h)\, dK = 0 \tag{3.18}$$

for all $\boldsymbol{v}_h \in \boldsymbol{V}_h$ and $q_h \in Q_h$.

Like in the simple example in Section 1.3 one can observe that the stabilization term is only non-zero when a residual $\mathcal{R}(\boldsymbol{u}_h)$ is present. If $\boldsymbol{u}_h$ approaches the exact solution the term vanishes. The choice of the stabilization parameter τ_{SUPG} is a research topic on its own and the interested reader is referred to [81] and the references therein. The given parameter is subsequently used

$$\tau_{SUPG} = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|\boldsymbol{u}_h|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-\frac{1}{2}} \tag{3.19}$$

and includes the chosen time step Δt and the element diameter h.

Remark 3. This stabilization is not assumed to be applied by default. It is in the following always mentioned, when the stabilized version is used and why it is appropriate in that case.

Another related stabilization is the Pressure Stabilizing Petrov Galerkin method introduced in [45]. Reviewing this approach, it allows the use of equal order elements and prevents the restriction from the inf-sup condition. Similar to the pressure defined outlet boundary condition, this leads to an unwanted effect in the discretized system that will be mentioned in the next paragraph and therefore this technique will not be considered further. Solving the discretized system in Problem 3.2 requires a nonlinear solution method like the Newton method to be used. A Newton method is based on the evaluation of the residual in the generically unknown vector x

$$F(x) = 0,$$

Algorithm 1 Newton's method

$x^0 \leftarrow$ initial guess
for $k = 1, 2, \ldots$ **do**
  Solve $J(x^{k-1})\delta x = -F(x^{k-1})$
  $x^k = x^{k-1} + \delta x$
end for

and a solution of a linear system at every iteration involving the system Jacobian

$$J(x) = \frac{\partial F(x)}{\partial x},$$

which resembles a directional derivative of the residual. The complete algorithm is given in Algorithm 1. Due to the large number of degrees of freedom involved in fluid flow based models, parallel computing techniques involving iterative solvers and preconditioners like Krylov methods are necessary. A comprehensive review of those techniques is given in [71]. For further analysis and discussion of the used techniques we consider the system of Problem 3.2. Applying Algorithm 1 we form the Jacobian by linearizing about a state x by applying the Gateaux derivative. This results in a block matrix of the form

$$J(x) = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}, \tag{3.20}$$

and is described as a saddle point system. The linear system, which has to be solved in every Newton iteration is now of the form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta x_p \end{bmatrix} = \begin{bmatrix} r_u \\ r_p \end{bmatrix}. \tag{3.21}$$

with r_u and r_p the residuals computed from $-F(x)$. These systems are notoriously hard to solve using Krylov methods like the Generalized minimal residual method (GMRES) and the effectiveness of the solver depends on the availability of a good preconditioner. While it is possible to define a preconditioner for the whole block matrix like domain decomposition techniques, one is advised to use specifically developed methods. A common technique is to consider the pressure Schur complement which reads

$$S = BA^{-1}B^T \tag{3.22}$$

and results from the full factorization of the Jacobian in a lower triangular, diagonal and upper triangular part

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B^T \\ 0 & -S \end{bmatrix}. \tag{3.23}$$

Since it is necessary to provide the inverse of S, one also has to compute the inverse of A which is almost completely dense and therefore these specialized techniques focus on finding a good enough approximation to $\tilde{S} \approx S$. A very efficient technique for approximation, the Least-squares commutator (LSC), is described in [25] and summarized in the next paragraph. This technique approximates the Jacobian of Problem 3.2 by a block-triangular system with

$$J \approx \begin{bmatrix} A & B^T \\ 0 & -\tilde{S} \end{bmatrix} \tag{3.24}$$

and solving the system by using the second row to compute x_p

$$\delta x_p = -\tilde{S}^{-1} r_p$$

and using the solution to compute δx_u with

$$\delta x_u = A^{-1}(r_u - B^T \delta x_p).$$

The solution procedure requires the computation of two inverses $\tilde{S}^{-1}$ and A^{-1}. Following the descriptions in [25], $\tilde{S}$ is approximated by

$$\tilde{S} = -(BM_u^{-1}B^T)^{-1}(BM_u^{-1}AM_u^{-1}B^T)(BM_u^{-1}B^T)^{-1}$$

with M_u^{-1} the inverse of the mass matrix of the velocity space. Moreover we now have to provide solver and preconditioner combinations for

$$P = BM_u^{-1}B^T \tag{3.25}$$

and M_u^{-1}. The authors in the reference state that the matrix resulting from Equation (3.25) resembles a scaled Poisson matrix and therefore an optimal preconditioner for this is an Algebraic Multigrid technique [71] and the mass matrix of the velocity space is preconditioned with a diagonal smoother. This preconditioning algorithm allows the parallel computation using iterative solvers.

3.4 Coupling of fluid flow and heat transfer

Encapsulating multiple physical phenomena such as fluid flow and accompanied heat transfer or fluid structure interaction in one simulation is a topic of recent research [48]. Techniques and discussions in this resource are of great interest for the research in this work because of the multiphysics character of the Boussinesq equation which is used to model air flow and heat transfer. For the solution one has to identify how much accuracy is needed for the purpose that the simulation has to fulfill and weigh up the resulting computational needs for one simulation or number of simulations needed for further analysis like optimization. Consequently one has to decide between the formulation of a coupled or decoupled problem.

3.4.1 Loose and tight coupling

The solution techniques for multiphysics problems can be simplified and split into two major approaches. Consider a system with multiple physics described by a residual formulation using a partial differential equation which is also dependent on time

$$F_1(x_1, x_2) = \frac{\partial x_1}{\partial t} - f_1(x_1, x_2) = 0, \tag{3.26a}$$

$$F_2(x_1, x_2) = \frac{\partial x_2}{\partial t} - f_2(x_1, x_2) = 0. \tag{3.26b}$$

A common approach is the operator splitting approach described in Algorithm 2, which is convenient because it looks independently at the different physical components by taking the known previous timestep of the second variable and solving the first equation and afterwards using the known first variable to solve the second equation. This technique is easy to implement and available software and

Algorithm 2 Operator splitting

$\{x_1(t_0), x_2(t_0)\} \leftarrow$ initial guess
for $k = 1, 2, \ldots$ **do**
  Solve for one timestep $\frac{\partial x_1}{\partial t} - f_1(x_1, x_2(t_{k-1})) = 0$
  Solve for one timestep $\frac{\partial x_2}{\partial t} - f_2(x_1(t_k), x_2) = 0$
end for

solution techniques for the individual physics can be used, although the solution is bound to first order accuracy in time due to the splitting approach unless an involved process of higher order splitting methods is introduced. Furthermore, any coupling terms that would appear in a formulation which was built from scratch to support multiphysics are basically neglected. Due to this we refer to this approach as loose coupling.

If we formulate the residual as a vector residual that reads

$$F(x_1, x_2) = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = 0, \tag{3.27}$$

we can use a Newton method to couple the system using Algorithm 1. The authors in [48] also highlight that using the Newton method, thus building the Jacobian

$$\frac{\partial F(x)}{\partial x} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} \end{bmatrix} \tag{3.28}$$

in every iteration, includes the off diagonal terms. In a loose coupling approach these terms are not present and therefore the Newton method is referred to as being tightly coupled.

3.4.2 Preconditioning

Solving tightly coupled systems can be involving especially when tackling large problems which require Krylov methods and thus appropriate preconditioners. Recently there is a momentum in software packages to support physics based preconditioners or also called block preconditioners. To demonstrate the application we consider the Boussinesq equations Equation (3.5) which can be described with a combined residual

$$F(\boldsymbol{u}, p, T) = \begin{bmatrix} F_1(\boldsymbol{u}, p, T) \\ F_2(\boldsymbol{u}, p, T) \\ F_3(\boldsymbol{u}, p, T) \end{bmatrix} = 0,$$

and the corresponding Jacobian after discretization and linearization

$$\frac{\partial F(\boldsymbol{u}, p, T)}{\partial(\boldsymbol{u}, p, T)} = \begin{bmatrix} \frac{\partial F_1}{\partial u} & \frac{\partial F_1}{\partial p} & \frac{\partial F_1}{\partial T} \\ \frac{\partial F_2}{\partial u} & \frac{\partial F_2}{\partial p} & \frac{\partial F_2}{\partial T} \\ \frac{\partial F_3}{\partial u} & \frac{\partial F_3}{\partial p} & \frac{\partial F_3}{\partial T} \end{bmatrix} = \begin{bmatrix} A & B^T & C \\ B & 0 & 0 \\ D & 0 & E \end{bmatrix}. \tag{3.29}$$

Equation (3.29) clearly shows the physics coupling terms on the off-diagonal of the Jacobian. Finding a good preconditioner is a matter of approximating the inverse of an operator as accurately as necessary and efficiently as possible. Therefore we concentrate on the block diagonal part of Equation (3.29) which can be decoupled by ignoring the off-diagonal terms and formulating separate preconditioners for the first block of the variables $\boldsymbol{u}$ and p

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$$

and the advection-diffusion block of the variable T denoted by matrix E. As we saw in Section 3.3 for the velocity/pressure block it is now convenient to use the LSC technique. Due to the characteristics of the temperature block, we conveniently use an Algebraic Multigrid technique as well.

Remark 4. The reader has to keep in mind, that we just ignore the off-diagonal blocks in the preconditioning techniques. This does not degrade the tight coupling, since we still involve these blocks in the fully coupled Newton solve and their contributions to the overall residual are still present.

3.4.3 Numerical comparison

In this section we analyze the behavior of a natural convection problem described by Equation (3.5) with real physical parameters by using a loose and a tight

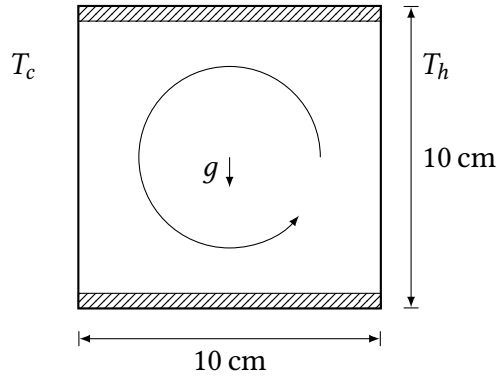


Figure 3.2: Natural convection problem in a cavity.

coupling approach to determine which formulation is appropriate for further algorithm development and suitability for optimization. We consider a two dimensional domain consisting of a square cavity with 10 cm side length and filled with air (c.f. Figure 3.2. The left side is fixed to the cold temperature of 293 K and the right side is fixed to the hot temperature of 294 K while the top and bottom walls are ideally insulated. We use a time dependent formulation (although this test is often done using an equilibrium assumption) to compare the time dependent behaviors of the coupling approaches. The density of air at around 293 K is $\rho = 1.225\,\mathrm{kg/m^3}$, the kinematic viscosity $\nu = 1.567 \times 10^{-5}\,\mathrm{m^2/s}$, the thermal conductivity $k = 0.025\,37\,\mathrm{W/(m\,K)}$, the specific heat capacity $C_p = 1005\,\mathrm{J/(kg\,K)}$ and we assume the thermal expansion coefficient to be $\alpha = 3.47 \times 10^{-3}\,1/\mathrm{K}$.

The loose coupling follows the regime of Algorithm 2, where the momentum equation of the Navier-Stokes equations is solved first using the temperature solution of the previous solve and then the computed velocity is used to compute the solution of the energy equation to determine the new temperature. Tight coupling is achieved by formulating a combined residual and using a monolithic Newton approach as shown in Algorithm 1.

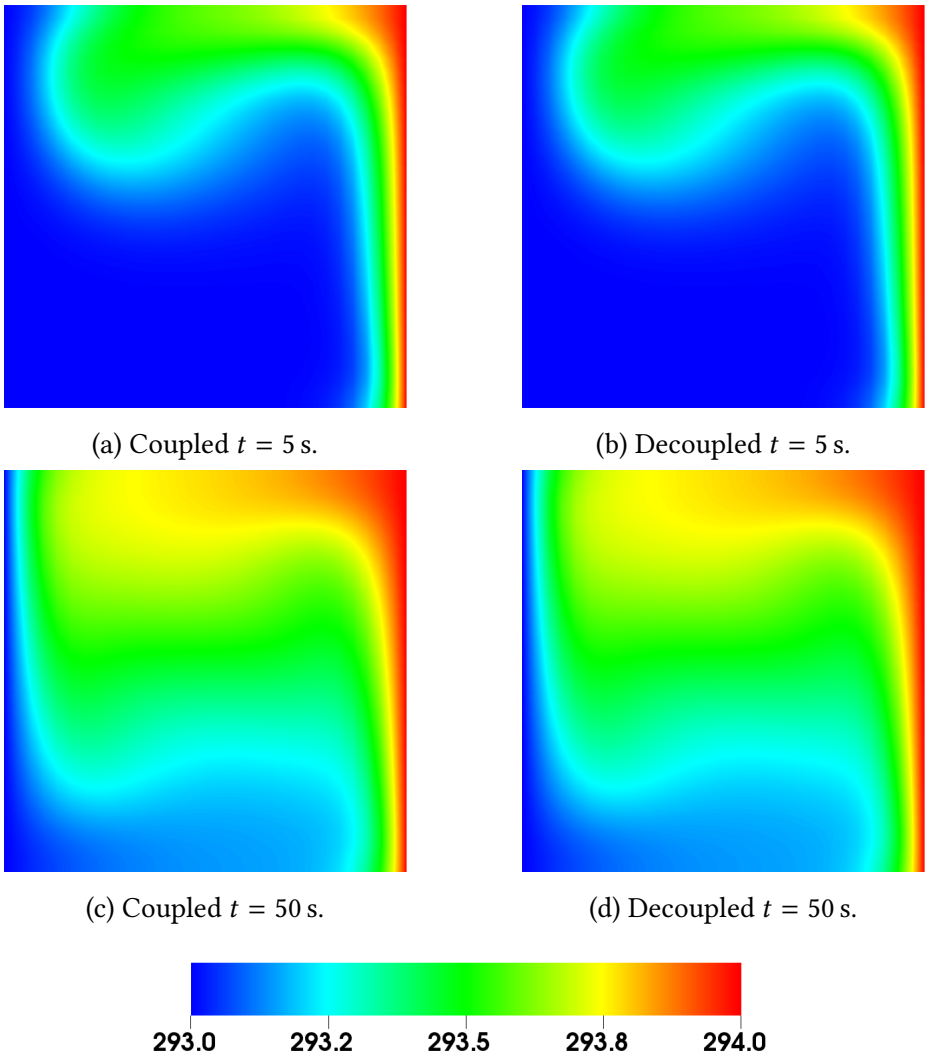
To determine a measure of correctness for comparison of both approaches, we assume that the tightly coupled method is "more correct" than the loosely coupled. Therefore we use the solution as a desired solution $x^* = [\boldsymbol{u}^*, p^*, T^*]^T$ to compute the infinity norm

$$\|x - x^*\|_\infty = \max_i |x(t_i) - x^*(t_i)|$$

and observe if or how much the solution deviates. One can see the visualized results in Figure 3.3 and we compute the previously discussed norm solely on the temperature field

$$\|T - T^*\|_\infty = 2.14 \times 10^{-8}. \tag{3.30}$$

Concluding from the result of Equation (3.30) and the visual solutions in Figure 3.3 there is just a slight numerical difference in the solution when comparing the two methods. There is no significant difference in computational cost and both simulations took ≈ 2.5 min to complete. This behavior leads to the assumption that it is safe to decouple the energy equation part in the solution of the Boussinesq equations if one does not strive for more than first order accuracy in time. It also encourages the intention in Chapter 4 where we try to decouple multiphysics to reduce computational in optimization problems.



(a) Coupled $t = 5$ s.

(b) Decoupled $t = 5$ s.

(c) Coupled $t = 50$ s.

(d) Decoupled $t = 50$ s.

(e) Color bar for the temperature profile range.

Figure 3.3: Comparison of loose and tight coupling in a natural convection problem. The temperature profile tightly coupled method is shown in the left column and the loose coupling in the right column.

3.5 Validation and verification

3.5.1 Flow around a cylinder

Besides the usual Method of Manufactured Solutions we test the developed code against a benchmark problem of a flow around a cylinder first described by [73]. Consider the non-dimensionalized incompressible Navier-Stokes Equations (3.3a) and (3.3b) with density $\rho = 1$ and kinematic viscosity $\nu = 0.001$. The geometry for this example is depicted in Figure 3.4 and consists of a channel $\Omega \in [0, 2.2] \times [0, 0.41]$ with a cylinder located at $(0.2, 0.2)$ with radius 0.05. The upper and lower wall as well as the cylinder boundary are modeled as walls with a no-slip boundary condition. The inflow boundary condition on the left is implemented with the profile

$$\boldsymbol{u} = \left[4u_m y(0.41 - y)0.41^2, 0\right]^T$$

and $u_m = 1.5$. The outflow boundary condition is implemented using a stress-free outflow described by

$$\nu \boldsymbol{u} \cdot \boldsymbol{n} - p\boldsymbol{n} = 0.$$

From these quantities and the characteristic length of the flow $L = 0.1$ we can compute the Reynolds number as

$$Re = \frac{u_m L}{\nu} = \frac{1.0 \cdot 0.1}{0.001} = 100.$$

To compare our results to the published benchmark results in [73] we compute the numerical quantities including the drag and lift forces described by the normal forces on the cylinder boundary by

$$[F_D F_L]^T = \int_{\Gamma_c} (\nu \nabla \boldsymbol{u} - p\mathbb{I})\boldsymbol{n}\, d\Gamma,$$

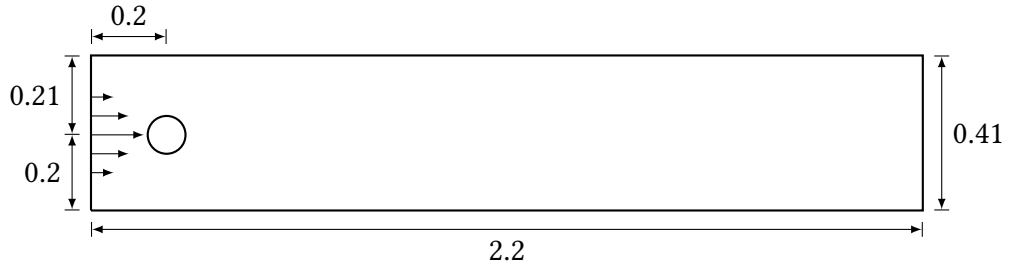


Figure 3.4: Flow around a cylinder.

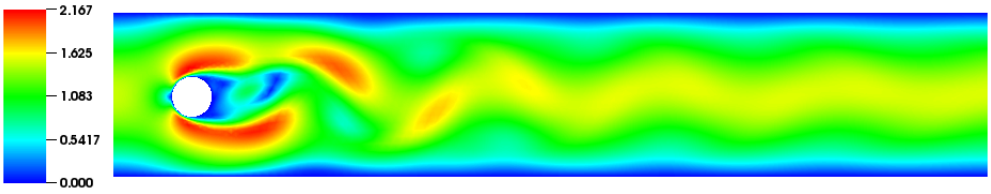


Figure 3.5: Velocity magnitude of the DFG 2D benchmark test of transient flow around a cylinder with fully developed periodic flow.

	$C_{D_{max}}$	$C_{L_{max}}$	P_d
Ref. [73]	[3.0804, 3.32]	[0.688, 1.25]	[2.0954, 2.6066]
Impl.	3.1161	1.0517	2.4458

Table 3.1: Comparison of numerical quantities with reference values from [73] for the DFG 2D benchmark test of transient flow around a cylinder.

and the corresponding drag and lift coefficients

$$C_D = \frac{2}{u_m^2 L} F_D, \quad C_L = \frac{2}{u_m^2 L} F_L.$$

Furthermore we also compute a pressure difference between the points $P_1 = (0.15, 0.2)$ and $P_2 = (0.25, 0.2)$ with

$$P_d = P_1 - P_2.$$

The simulation process matches the one described in [73], where the simulation is run until 25 s and an arbitrary cycle of the periodic flow with frequency $f = 0.33$ s is post processed to compute the maximum of C_D, C_L and mean of the pressure difference P_d. A snapshot of the velocity magnitude profile in that time frame can be seen in Figure 3.5.

The authors in [73] gathered results from a variety of participants in the benchmark study and provide reference values for the computed quantities. There is no specific participant in the reference that used a comparable algorithm to the one suggested in this work for the solution above. The closest example is a similar discretization using a compatible Taylor-Hood finite element pair $(\mathcal{P}_2/\mathcal{P}_1)$, but with a segregated or pressure projection scheme. Hence the three most spatially refined results of each participant are used for a comparison range in Table 3.1. The simulation was run with a constant time step of $\Delta t = 0.005$ using the BDF-1 method and a spatial discretization with 28127 degrees of freedom. As one can see, all computed values lie within the reference range.

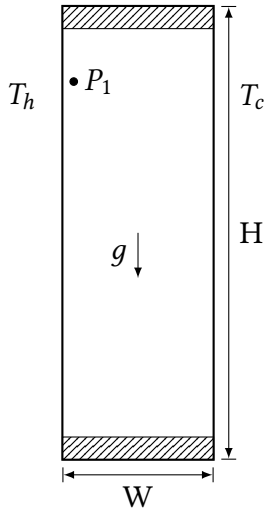


Figure 3.6: Cavity.

3.5.2 MIT cavity benchmark

The MIT cavity benchmark example proposed by [17] simulates a flow in a cavity driven by a temperature difference of the side walls modeled by the Boussinesq Equation (3.5). In this particular case Equation (3.5) is non dimensionalized

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \sqrt{\frac{Pr}{Ra}}\nabla^2 \boldsymbol{u} - \boldsymbol{g}T \qquad \text{in } \Omega \times \mathbb{R}, \tag{3.31a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \times \mathbb{R}, \tag{3.31b}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T = \frac{1}{\sqrt{RaPr}}\nabla^2 T \qquad \text{in } \Omega \times \mathbb{R} \tag{3.31c}$$

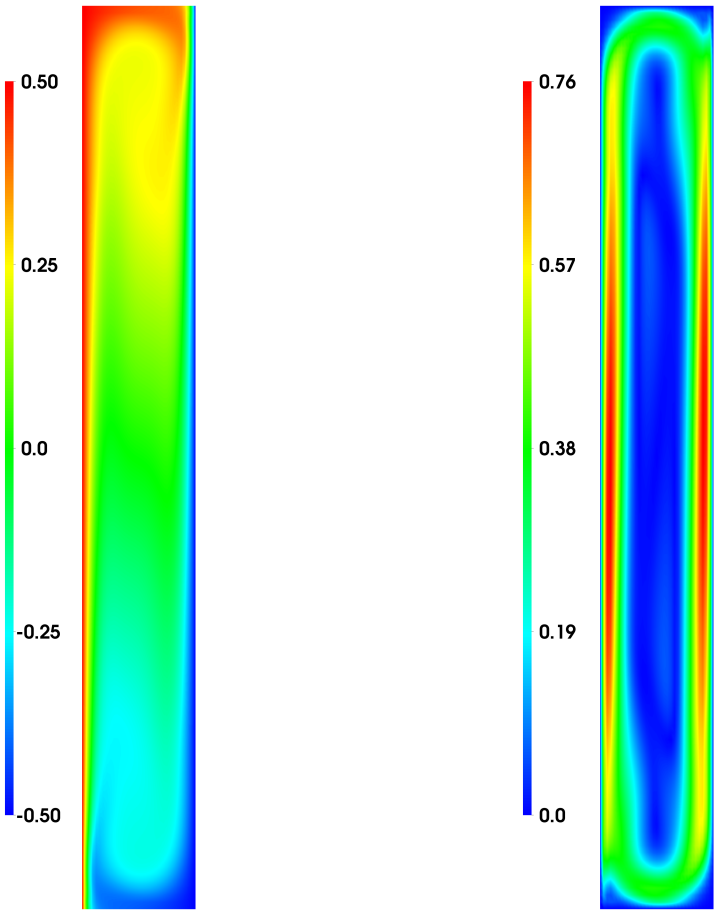
and described by the values of the Rayleigh number $Ra = 3.4 \times 10^5$ and the Prandtl number $Pr = 0.71$. The height of the cavity is set with $H = 8$ and the width is $W = 1$. All velocity boundary conditions are chosen to be no-slip on the walls and a hot temperature $T_h = 0.5$ on the left as well as a cold temperature $T_c = -0.5$ on the right are introduced. The top and bottom of the cavity are assumed to be isolated and gravity $\boldsymbol{g}$ is pointing downwards. An illustration of this can be viewed in Figure 3.6. The objective described in [17] is to compute the quantities of the velocity $\boldsymbol{u}$ and the temperature T at different points in the domain (although just $P_1 = (0.1810, 7.3700)$ is considered here) as well as the Nusselt number

$$Nu = \frac{1}{H}\int_0^H \frac{\partial T}{\partial x}|_{x=0,W}\, dy$$

	u_1	T_1	$-Nu$
⋆	0.058	0.26	4.59
Turek	0.057	0.26	4.58
Davis	0.056	0.27	4.58
Gresho	0.057	0.27	4.58
Le Quéré	0.056	0.27	4.58

Table 3.2: Computed quantities of the MIT benchmark of a heated cavity with 8:1 aspect ratio at measurement point P_1 (denoted by ⋆) compared to results gathered in [21].

on the walls. The spatial discretization uses a structured grid built with 40 triangles in the horizontal and 100 triangles in the vertical direction. We use no local refinement like other authors mentioned in [21] do. Using the mentioned discretization this results in 25 754 degrees of freedom in the nonlinear system which is not nearly as much as the numbers mentioned in the publication with a range from ∼ 85 000 to ∼ 560 000. In [21] a few results are gathered from different authors and listed for comparison in Table 3.2. The temperature and velocity profile with fully developed oscillating flow can be viewed in Figure 3.7. Although there may be small differences, the results are conforming with the gathered results of other authors. Reviewing the comparison, one has to keep in mind the difference in grid resolution and choice of element types (quadrilaterals in most references and triangles in the present implementation) as well as solution and preconditioning method of the linear and nonlinear system.



(a) Temperature.

(b) Velocity magnitude.

Figure 3.7: MIT cavity benchmark example showing the temperature profile in (a) and the velocity magnitude in (b).

Chapter 4

Optimal control

One of the main aspects in this work is to investigate viable methods for optimization problems in multiphysics systems using the finite element method. Looking at the examples developed in Chapter 2 and Chapter 3 we can define situations in which optimal control approaches are reasonable. Taking for example a heating process of a plate where the objective is to achieve a certain homogeneous temperature at a specific time on the material controlled by heating elements. The problem here would be compute a time dependent trajectory for the heating elements that fulfill these conditions and upholds the underlying physics. This can be described in the general context of PDE constrained optimization. In most literature (see i.e. [83]) these problems are defined on decision variables x for the optimal control problem which are in the function space of the partial differential equation and therefore $x \in H^1$ or $x \in L^2$. In all following cases we only consider decision variables $x \in \mathbb{R}^n$. This is achieved by mapping operations that describe for example a percentage or amplitude of an actuator that heats a certain part of a boundary. Details of this process are further explained in the upcoming sections.

In recent publications many different techniques have evolved regarding the control of heating ventilation and air conditioning (HVAC) systems. The most similar approach to the method presented in this work is [40], where the Navier-Stokes equations are used in combination with a diffusion-convection model but a stationary air flow profile is assumed and therefore no time dependency in the Navier-Stokes equations is introduced. Furthermore in [40] the process involves the full set of equations in the optimization process. Similarly the authors in [11, 16] analyzed a linear-quadratic regulator approach based on the linearized Navier-Stokes model combined with stationary air flow profiles. Other approaches like the one described in [38, 39] use similar techniques based on lower dimensional ordinary differential equation models and using Model Predictive Control methods. These methods are generally more computationally efficient, but cannot make use of the air flow profile information.

4.1 PDE constrained optimization

With the intention to solve the mentioned tasks, like finding an input for a heating problem that suffices certain conditions and restrictions, one has to properly define an optimal control problem in a suitable mathematical setting. A general optimization problem can be described as follows:

Problem 4.1 (General optimization problem). *Let $x \in \mathbb{R}^n$ and f, g_i, h_j be functions, then we can formulate a static optimization problem by*

$$\min_x \quad f(x) \tag{4.1a}$$

subject to

$$h_i = 0, \tag{4.1b}$$

$$g_i \leq 0, \tag{4.1c}$$

$$x \in [x_l, x_u], \tag{4.1d}$$

with g_i the, probably nonlinear, functions describing inequality constraints and h_i describing equality constraints. Equation (4.1d) *prescribes the bounds that the decision variables x are allowed to reside in, with x_l and x_u being lower and upper bounds respectively. The objective is now to find and a solution for x that minimizes $f(x)$ such that all constraints are fulfilled.*

To motivate the discussion in the next sections an example of an optimal control problem for a partial differential equation is developed. These presented concepts include only a brief overview of the existing theory and are required to explain the background in the process of the next sections and concepts from [83, 36] are used. Looking at a simplified example of Equation (2.1) we define the stationary heat equation with density $\rho = 1$, specific heat capacity $C_p = 1$ and thermal conductivity $k = 1$ as

$$-\nabla^2 T = 0 \qquad \text{in } \Omega, \tag{4.2a}$$

$$\nabla T \cdot \boldsymbol{n} = \alpha(Q - T) \qquad \text{on } \Gamma. \tag{4.2b}$$

Furthermore we define a so-called objective functional where we assume that Q is our input. A typical objective functional is the quadratic term

$$J(T, c) = \frac{1}{2}\|T - T^*\|^2_{L_2(\Omega)} + \frac{\gamma}{2}\|Q\|^2_{L_2(\Gamma)} \tag{4.3}$$

which states that the solution T should be as close to the desired solution T^* as possible in the L_2 norm while keeping the control input Q small. The variable γ serves as a tunable Tikhonov regularization parameter.

To understand the necessity of a mapping for the control variable mentioned in the introduction of this chapter, one has to look at the physical aspects of the terms. When $Q \in L_2$ is the control variable for our problem, then this means that we have a physical device that can heat infinitely small parts of a domain to a specific temperature (and therefore in L_2). Here, available devices work with different concepts and provide a constant temperature for the whole actuator device and thus heat homogeneously on the whole domain of the device and therefore the boundary part of the domain. Although it might be viable to look at the problems with a control input defined in L_2, one has to adapt the solution or hardware afterwards in order to comply with a physical example. Concluding these thoughts a simple mapping is introduced by using the decision variable $c \in \mathbb{R}$, which represents a percentage of a possible control range for an input $Q \in L_2$ such that the problem in Equation (4.2) is reformulated as

$$-\nabla^2 T = 0 \qquad \text{in } \Omega, \tag{4.4a}$$

$$\nabla T \cdot \boldsymbol{n} = \alpha(cQ - T) \qquad \text{on } \Gamma. \tag{4.4b}$$

The corresponding objective functional is then

$$J(T, c) = \frac{1}{2}\|T - T^*\|^2_{L_2(\Omega)} + \frac{\gamma}{2}c^2. \tag{4.5}$$

This formulation restricts the solution space numerically in comparison to an L_2 control, since Q is constant, but represents a more physical formulation for the actuation. This mapping is feasible, if $Q \in L_2$ and $|c| < \infty$ then it holds that $cQ \in L_2$ since

$$\|Q\|_{L_2} = \left(\int_\Omega |Q|^2 \, dx \right)^{1/2} < \infty, \tag{4.6}$$

then

$$\|cQ\|_{L_2} = \left(\int_\Omega |cQ|^2 \, dx \right)^{1/2} < \infty. \tag{4.7}$$

The objective depends on two unknown variables, the temperature T and the control input c. In this context it is assumed that the primary state variable (in this case T) always has to be computed first and thus it is possible to write $J(c)$. Finally, we state an optimal control problem, which is constrained by a partial

differential equation using the formulation

$$\min_{c} \quad J(c) = \frac{1}{2}\|T - T^*\|^2_{L_2(\Omega)} + \frac{\gamma}{2}c^2 \tag{4.8a}$$

subject to

$$F(T) = 0, \tag{4.8b}$$

$$c \in [0, 1]\,, \tag{4.8c}$$

where $F(T)$ is the residual formulation of Equation (4.4). To solve an optimal control problem defined by Equation (4.8) the Karush-Kuhn-Tucker (KKT) conditions have to be fulfilled. These define first-order necessary conditions for the solution to be optimal [12]. They state that $\bar{x}$ is an optimal point of Problem 4.1 and it fulfills the following conditions

$$\nabla f(\bar{x}) + \sum_{i=1}^{m} \mu_i \nabla g_i(\bar{x}) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\bar{x}) = 0 \tag{4.9a}$$

$$g_i(\bar{x}) \leq 0 \qquad \text{for all } i = 1, \ldots, m \tag{4.9b}$$

$$h_j(\bar{x}) = 0 \qquad \text{for all } j = 1, \ldots, l \tag{4.9c}$$

$$\mu_i \geq 0 \qquad \text{for all } i = 1, \ldots, m \tag{4.9d}$$

$$\mu_i g_i(\bar{x}) = 0 \qquad \text{for all } i = 1, \ldots, m. \tag{4.9e}$$

The first step in solving the example in Equation (4.8) is to define a Lagrangian (the KKT conditions are a generalization of Lagrange multipliers [12, 53]) by taking the inner product with the Lagrange multiplier λ, defined in an appropriate space, with the equality constrained partial differential equation and subtracting it to the objective functional we get

$$\mathcal{L} = J(c) + \int_{\Omega} \nabla^2 T \lambda \, dx. \tag{4.10}$$

Remark 5. The reader should note, that the formulation for Hilbert spaces is used and thus an inner product notation is used.

To fulfill Equation (4.9) the gradient of the Lagrangian has to be computed which is done by taking the Gateaux derivative of Equation (4.10). This is usually a tedious process even for the most simple problems and the next part should gather as much steps as possible in order to cover the common pitfalls in the process. Therefore the control variable c has to be perturbed by $h\hat{c}$. This operation basically means that we analyze what a small difference in the control input would have on the system and thus the output, hence this is also called the sensitivity. If a variable is dependent on the perturbation the short notation

$$\tilde{T} = T(c + h\hat{c})$$

is used. The Gateaux derivative of Equation (4.10) reads

$$\delta\mathcal{L} = \frac{\partial}{\partial h}\left[\int_\Omega \frac{1}{2}(\tilde{T} - T^*)^2\,dx + \frac{\gamma}{2}(c + h\hat{c})^2 + \int_\Omega \lambda\nabla^2\tilde{T}\,dx\right]_{h=0}.$$

Differentiating the first and second term and substituting

$$\hat{T} = \frac{\partial}{\partial h}\Big[T(c + h\hat{c})\Big]_{h=0},$$

as well as using the chain rule for the first term where it holds that

$$\frac{\partial}{\partial h}\left[\int_\Omega \frac{1}{2}(T(c + h\hat{c}) - T^*)^2\,dx\right]_{h=0} = \int_\Omega \left.\frac{\partial T(c + h\hat{c})}{\partial h}\right|_{h=0} (T(c + h\hat{c})|_{h=0} - T^*)\,dx$$

this leads to

$$\delta\mathcal{L} = \int_\Omega (T - T^*)\hat{T}\,dx + \gamma c\hat{c} + \frac{\partial}{\partial h}\left[\int_\Omega \lambda\nabla^2\tilde{T}\,dx\right]_{h=0}. \tag{4.11}$$

Using partial integration and Greens identity for the last term twice results in

$$\begin{aligned}\delta\mathcal{L} = &\int_\Omega (T - T^*)\hat{T}\,dx + \gamma c\hat{c} \\ &+ \frac{\partial}{\partial h}\left[\int_\Omega \tilde{T}\nabla^2\lambda\,dx - \int_\Gamma \tilde{T}\nabla\lambda\cdot\boldsymbol{n}\,ds + \int_\Gamma \lambda\nabla\tilde{T}\cdot\boldsymbol{n}\,ds\right]_{h=0}\end{aligned} \tag{4.12}$$

Substituting the boundary condition like in the process of a variational formulation in the last term yields

$$\begin{aligned}\delta\mathcal{L} = &\int_\Omega (T - T^*)\hat{T}\,dx + \gamma c\hat{c} \\ &+ \frac{\partial}{\partial h}\left[\int_\Omega \tilde{T}\nabla^2\lambda\,dx - \int_\Gamma \tilde{T}\nabla\lambda\cdot\boldsymbol{n}\,ds\right]_{h=0} \\ &+ \frac{\partial}{\partial h}\left[\int_\Gamma \lambda(-\alpha\tilde{T} + \alpha Q(c + h\hat{c}))\,ds\right]_{h=0}.\end{aligned} \tag{4.13}$$

Now by pulling in the derivative $\frac{\partial}{\partial h}$ and setting $h = 0$ afterwards

$$\begin{aligned}\delta\mathcal{L} = &\int_\Omega (T - T^*)\hat{T}\,dx + \gamma c\hat{c} + \int_\Omega \hat{T}\nabla^2\lambda\,dx \\ &- \int_\Gamma \hat{T}\nabla\lambda\cdot\boldsymbol{n}\,ds - \int_\Gamma \alpha\lambda\hat{T}\,ds + \int_\Gamma \alpha\lambda Q\hat{c}\,ds.\end{aligned} \tag{4.14}$$

Finally, terms containing $\hat{T}$ and $\hat{c}$ are collected to visually differentiate between the two computed sensitivities

$$\delta\mathcal{L} = \int_\Omega (T - T^*)\hat{T}\,dx + \int_\Omega \nabla^2\lambda\hat{T}\,dx - \int_\Gamma \hat{T}\nabla\lambda\cdot\boldsymbol{n}\,ds - \int_\Gamma \alpha\lambda\hat{T}\,ds + \gamma c\hat{c} + \int_\Gamma \alpha Q\lambda\hat{c}\,ds. \tag{4.15}$$

The so called adjoint equations are obtained from the $\hat{T}$ terms, which is stated in residual form, but can easily be reformulated as a partial differential equation with boundary conditions like

$$-\nabla^2\lambda = T - T^* \qquad \text{in } \Omega, \tag{4.16a}$$

$$\nabla\lambda\cdot\boldsymbol{n} + \alpha\lambda = 0 \qquad \text{on } \Gamma. \tag{4.16b}$$

Equation (4.16) does not depend on the control input c and is only driven by the $T - T^*$, the difference from the state and the desired state. Furthermore a reduced gradient can be obtained by assuming Equation (4.16) can be solved and thus the residual appearing in the Lagrangian is zero and yields by dividing with $\hat{c}$

$$\frac{\partial J}{\partial c} = \gamma c + \int_\Gamma \alpha Q\lambda\,ds. \tag{4.17}$$

The presented derivation is a classical way of first optimize then discretize. An alternative way, called first discretize then optimize, is discussed in the context of gradient computation in the next section.

Remark 6. Note that this derivation could also be done in the way that Q would be perturbed by $h\hat{Q}$ and afterwards is defined to be dependent on c. The notation of $Q(c)$ would be introduced and the derivation would be the same and still correct, since this is only a matter of preference and notation to make clear that the control and thus the gradient are in $\mathbb{R}$.

Reviewing Equation (4.9) and assuming no inequality constraints ($m = 0$) the state equation Equation (4.4) and adjoint Equation (4.16) fulfill the KKT conditions if the control input $\bar{c}$ is in the admissible set defined by the bounds. An intuitive way of solving the problem is to formulate them as a monolithic system, which resembles the full space method, where all variables are solved simultaneously for example with a Newton method. This results in a complex block system which is often ill-conditioned (see e.g. [1]). In time dependent problems the block system increases to unmanageable dimensions and one is forced to use fixed time grids. Due to the problems here being time dependent, this method is not considered and we concentrate on the reduced space approach in which

Algorithm 3 Reduced space optimization

$c^0 \leftarrow$ initial guess
for $k = 1, 2, \ldots$ **do**
  Solve state equation (4.4) for T
  Solve adjoint equation (4.16) for λ
  Compute $\frac{\partial J}{\partial c} = \gamma c + \int_\Gamma \alpha Q \lambda \, ds$
  Decide how to change c and continue
end for

the dependent variables are basically eliminated by solving the state and adjoint equations separately. A general way for solving would be to first compute a solution of the state equation Equation (4.4) and therefore eliminate the unknown T from Equation (4.16). Afterwards the adjoint equation is solved to obtain λ. In every case, this system is linear and depends on the state of the state equation. Finally the gradient Equation (4.17) can be computed using the adjoint variable λ. The procedure is summarized in Algorithm 3. This method is attractive because it allows the decoupling of the solution of the state and adjoint equation and it is possible to use specific numerical techniques for each problem. There exist a variety of solution methods to solve optimization problems as the one defined above and an overview is given in [47]. The most common are gradient and Newton or Quasi-Newton type methods. For the sake of simplicity these methods are assumed to be given in the following context.

Remark 7. Since this work does not deal with decision variables in spaces other than $\mathbb{R}$ it is possible to utilize given implementations of the mentioned optimization algorithms. If the decision variables is defined in a vector function space like H^1 or L_2 most of the implementations do not comply with the inner product choices and have to be used very carefully. One should review this problem in [77] where a detailed explanation is given regarding the problem.

Remark 8. In this remark a brief overview about the two fundamental methods on how to derive an expression for the gradient is given. As introduced in the last section the computation of the gradient is an elementary step in solving an optimization problem. The presented approach leaves the equation as is and starts to derive a formulation that optimizes the problem. For numerical computation the results still have to be discretized in space and often also in time. Thus it is called first optimize then discretize and allows a thorough breakdown on a mathematical basis during the whole derivation but requires an involving process to compute the adjoint equation and gradient expression symbolically by hand. Further benefits are that one is able to theoretically use different discretization methods for the state and adjoint equation since the formulation does not assume

anything specific. Even time discretization can be chosen completely independent and an implementation of the state equation using backward Euler and a forward Euler method for the adjoint is possible (assuming numerical stability). Hence it is even possible to use adaptive time stepping methods such that dynamic characteristics can be exploited and computational times can be greatly reduced (see e.g. [78]).

In the other major approach the state equation and objective functional are discretized in the first step and the Lagrangian is formed from that. Then optimality conditions are derived from the discretized equations. A considerable advantage to this method is the possibility to use automatic differentiation which allows us to derive the gradient of the Lagrangian by simply using the appropriate software calls in the implementation which is much less error prone than the derivation by hand. This leads to the aspect that the discretization in space and time has to be the same for the state and adjoint equation and there is no possibility known to the author to use adaptive time stepping in that context. As mentioned in [1] and the references therein, in particular [36], the optimality conditions can be very different from the first optimize then discretize approach but there are also references that show a corresponding gradient using automatic differentiation and a symbolic derivation of the adjoint [44]. Since the first optimize then discretize approach allows for more flexibility, the method is used in the following.

4.2 Heating process of an aluminum block

To demonstrate the application of an optimization problem on a physical example, a benchmark was developed in the course of this work. An illustration of the CAD model can be seen in Figure 4.1. The construction consists of a $15\,\mathrm{cm} \times 15\,\mathrm{cm} \times 10\,\mathrm{cm}$ aluminum block of EN AW 5083 material. The 36 side holes fit actuators with 8 mm diameter and 40 mm length. Not all holes are occupied by actuators but rather should provide options for different actuator configurations. The actuators are cartridge heaters from the company hotset GmbH, specifically hotrod type HHP. Sensors are placed on the surface for validation and are thermocouple elements of type J. In a benchmark run there are a total of 8 actuators and 8 sensors connected to a hotset `hotcontrol` cDT 12 which communicates with a `LabView` computer connected via Ethernet.

4.2.1 Mathematical modeling and parameter identification

At first the behavior of the actuators is analyzed. For this purpose we use the hotcontrol unit and put the desired temperature to 333.15 K and measure the temperature at the tip of the actuator with the integrated sensor. As one can

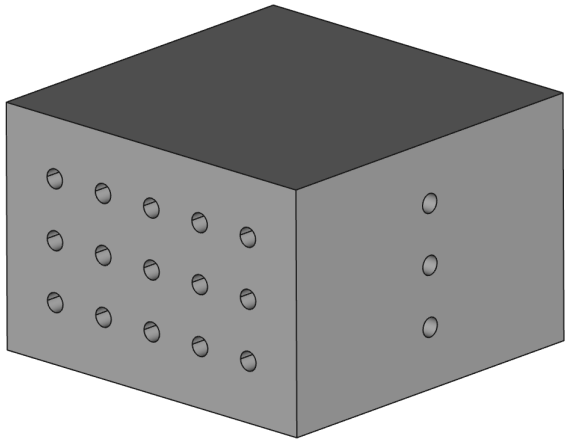


Figure 4.1: Illustration of the CAD model for the heating benchmark problem.

see in Figure 4.2, the temperature rises from 302 K to the desired temperature of 333.15 K in 244 s with a slight overshoot, which happens because of lower level controllers in the hotcontrol unit. Looking at the steep gradient in Figure 4.2 we can see the 30 K difference in the first 9 s where the actuator is active. Therefore we can verify the manufacturer response time of 0.3 s per K.

In order to compute appropriate optimization examples, the simulation model has to be defined and discretized properly. We define the computational domain with the aluminum block with Ω and the boundaries $\Gamma := \Gamma_s \cup \Gamma_c$ with contact to the actuators are defined by Γ_s, where s indicates an incoming heat flux from a source. Boundaries with no contact to another solid material are indicated with Γ_c (c for cooling). Furthermore a measurement boundary Γ_{top} at $y = 15\,\text{cm}$ in the XZ plane is defined. Then the model can be described by:

$$\rho C_p \frac{\partial T}{\partial t} - k\nabla^2 T = 0 \qquad \text{in } \Omega \times \mathbb{R}, \tag{4.18a}$$

$$\boldsymbol{n} \cdot k\nabla T = c_i Q_i \qquad \text{on } \Gamma_{s_i} \times \mathbb{R}, \tag{4.18b}$$

$$\boldsymbol{n} \cdot k\nabla T = h(T_\infty - T) \qquad \text{on } \Gamma_c \times \mathbb{R}. \tag{4.18c}$$

Before trying to compute a solution to an optimal control problem the physical material parameters have to be identified. For this purpose guideline variables were chosen from the materials as follows

- density $\rho = 2700\,\text{kg/m}^3$,

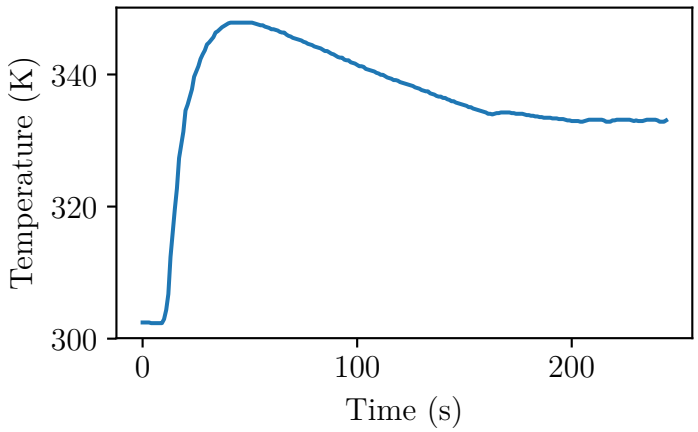


Figure 4.2: Actuator jump from 302 K to 333.15 K.

- specific heat capacity $C_p = 921\,\mathrm{J/kg\,K}$,
- thermal conductivity $k = 210\,\mathrm{W/m\,K}$.

The only missing parameter is the heat transfer coefficient h which in this case has to be defined for aluminum and air and no forced convection (no air flow assumed). According to [85] this parameter is in the range of $\approx 5\,\mathrm{W/m^2\,K}$ for a temperature difference of 30 K. Equation (4.18) is discretized using the finite element method described in Section 2.3 and different values for h are simulated and evaluated by comparison with on measurements. The `MFEM` implementation introduced in Section 2.3 was used for the identification because of the adaptive time stepping capabilities. The solutions are compared against a run on the

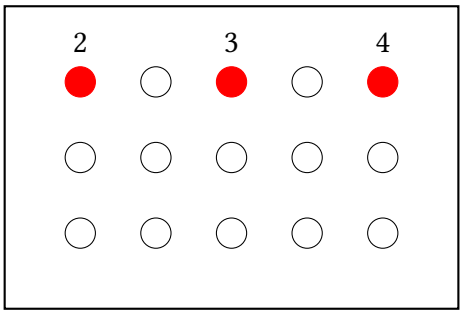


Figure 4.3: Actuator configuration from the front view in the XY plane.

physical benchmark that used 7 actuators and 9 sensors in the configuration seen in Figures 4.3 to 4.5 which heat the aluminum block from 301 K on the top surface to $\approx$ 380 K by setting a constant actuator percentage of 25% during a 9000 s time frame. The sensor configuration included 7 sensors on the top surface

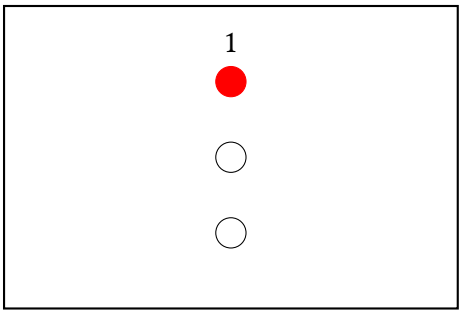


Figure 4.4: Actuator configuration from the side view in the YZ plane.

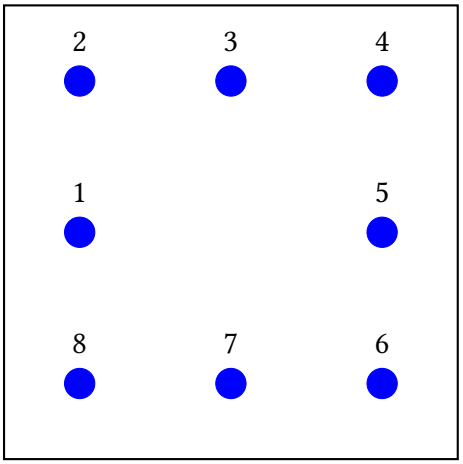


Figure 4.5: Sensor configuration from the top view in the XZ plane.

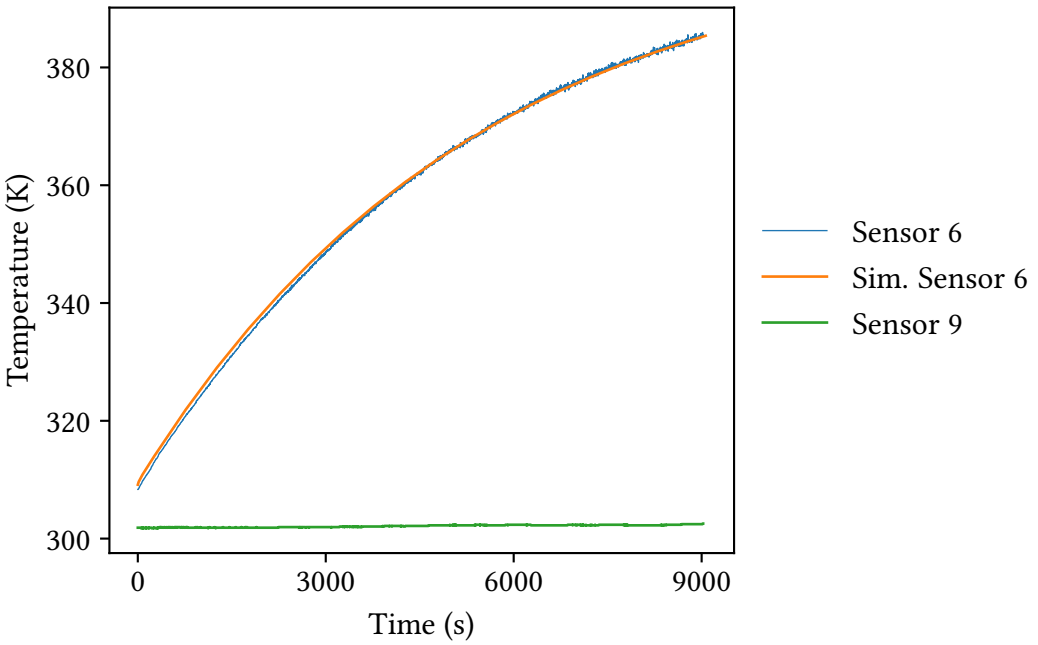


Figure 4.6: 25 % on 7 actuators during a 9000 s timeframe. Sensor 9 is an actuator in this case which is used only for measurement.

because one sensor was used to measure the ambient temperature to verify the hypotheses that the ambient temperature does not change significantly in a small distance (here $\approx$ 10 cm) of the block. Figure 4.6 shows the comparison using $h = 7.5\,\mathrm{W/m^2\,K}$. It is clear to see that the simulation matches the behavior of the physical benchmark and therefore is a suitable model for further analysis in optimal control problems.

4.2.2 Homogeneous temperature example

As an example for an optimal control problem that can be tested on the experiment consider the setup described in Section 4.2.1 so that the 8 actuators and 8 sensors are in the same positions as in Figures 4.3 to 4.5. An objective functional that penalizes the profile difference from the desired profile at a specific end time t_e as well as the control input amplitude c_i of each of the $N_a = 8$ actuators over the whole time frame is introduced. The complete problem is described by

$$\min_{c} \quad J(c) = \sum_{i=1}^{N_a} \int \frac{\gamma_1}{2} c_i^2 \, dt + \int_{\Gamma_{top}} \frac{\gamma_2}{2} (T(t_e) - T^*)^2 \, ds \tag{4.19a}$$

subject to

$$\rho C_p \frac{\partial T}{\partial t} - k\nabla^2 T = 0 \qquad \text{in } \Omega \times \mathbb{R}, \tag{4.19b}$$

$$\boldsymbol{n} \cdot k\nabla T = c_i Q_i \qquad \text{on } \Gamma_{s_i} \times \mathbb{R}, \tag{4.19c}$$

$$\boldsymbol{n} \cdot k\nabla T = h(T_\infty - T) \qquad \text{on } \Gamma_c \times \mathbb{R} \tag{4.19d}$$

and

$$-10 \leq \frac{\partial c_i}{\partial t} \leq 10 \qquad \text{for all } t \in (t_0, t_e), \tag{4.19e}$$

$$0 \leq c_i \leq 100 \qquad \text{for all } t \in (t_0, t_e). \tag{4.19f}$$

The equality constraint now contains the time dependent partial differential equation of the transient heat equation. Furthermore an inequality constraint is introduced in Equation (4.19e) whose purpose is to limit the rate of change of the control input $c_i(t)$ to not change more than 10% in every control interval. To work with a gradient based optimization algorithm the Lagrangian has to be formulated and an expression for the gradient has to be derived. As in previous examples the continuous adjoint method is used to compute the gradient and the process deriving the adjoint equation for the problem above can be reviewed in Appendix A. The adjoint equation then reads

$$-\rho C_p \frac{\partial \lambda}{\partial t} - k\nabla^2 \lambda = 0 \qquad \text{in } \Omega \times \mathbb{R}, \tag{4.20}$$

$$k\nabla\lambda \cdot \boldsymbol{n} = 0 \qquad \text{on } \Gamma_{s_i} \times \mathbb{R}, \tag{4.21}$$

$$k\nabla\lambda \cdot \boldsymbol{n} + \kappa\lambda = 0 \qquad \text{on } \Gamma_c \times \mathbb{R}. \tag{4.22}$$

It should be pointed out that the time differentiated part in the equation has a negative sign and therefore has to be time reversed. The negative term would go from t_0 to t_e, as derived in Appendix A and by changing the sign and thus switching the bounds of the time interval the time for the adjoint equation goes from t_e to t_0. The initial condition can be derived (since the equation is time reversed this is indicated with t_e) from

$$\rho C_p \lambda(t_e)|_{\Gamma_{top}} + \gamma_2(T(t_e) - T^*) = 0 \tag{4.23}$$

$$\lambda(t_e)|_{\Gamma_{top}} = -\frac{\gamma_2}{\rho C_p}(T(t_e) - T^*) \tag{4.24}$$

and the gradient expression is given by

$$\frac{d\mathcal{L}_i}{dc_i} = \int \gamma_1 c_i \, dt - \iint_{\Gamma_{s_i}} \lambda Q_i \, ds dt. \tag{4.25}$$

The solving process requires Equations (4.19b) and (4.20) to be discretized in space and time. In this case the finite element method for discretization in space and the BDF-1 method with equidistant time steps is used for discretization in time. Every solution of the state equation is saved temporarily so that no snapshotting and interpolation algorithm needs to be implemented to have the states available for the adjoint solve. These choices were made to have a simple proof of concept configuration in the `Firedrake` environment. The sequential quadratic programming algorithm for large scale systems implemented in `SNOPT` [32] is used to solve the numerical optimization problem. Since the problem includes inequality constraints, the optimization algorithm implementation in `SNOPT` requires a Jacobian matrix for these. By discretizing Equation (4.19e) with first order finite differences

$$g(c) = \frac{\partial c}{\partial t} \approx \frac{c^{n+1} - c^n}{\Delta t}$$

it is straight forward to compute the Jacobian

$$\frac{\partial g(c)}{\partial c} = \begin{bmatrix} -\frac{1}{\Delta t} & \frac{1}{\Delta t} & 0 & \dots & 0 \\ 0 & -\frac{1}{\Delta t} & \frac{1}{\Delta t} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -\frac{1}{\Delta t} & \frac{1}{\Delta t} \end{bmatrix}.$$

The ambient temperature T_∞ was measured at the time of computation as 301.15 K and set accordingly with the heat transfer coefficient of $h = 7.5\,\mathrm{W/m^2\,K}$. Regularization parameters were set to $\gamma_0 = 0.0$, $\gamma_1 = 1 \times 10^2$ and $\gamma_2 = 1 \times 10^{-6}$. The desired temperature is set to $T^*|_{\Gamma_{top}} = 303.15\,\mathrm{K}$ which has to reached in the time frame from $t_0 = 0\,\mathrm{s}$ to the final time $t_e = 30\,\mathrm{s}$. The major optimization tolerance (termination criterion) for `SNOPT` was chosen to 1×10^{-4}. Since the chosen time step is equal for both state and control input the number of decision variables can be computed by the number of actuators multiplied by the number of time steps

$$N_a \times N_t = 240. \tag{4.26}$$

Running the computation, the implementation using `SNOPT` is able to solve the problem in 5 major iterations taking 52.42 s. The functional can be reduced from an initial value of $J = 0.8$ to a final value of $J = 9.2 \times 10^{-5}$.

Remark 9. Since `SNOPT` is a serial implementation of an optimization algorithm, only the state and adjoint equations are solved in parallel. The optimization process and algorithm runs on the root process and communicates the necessary data to the other processes and vice versa.

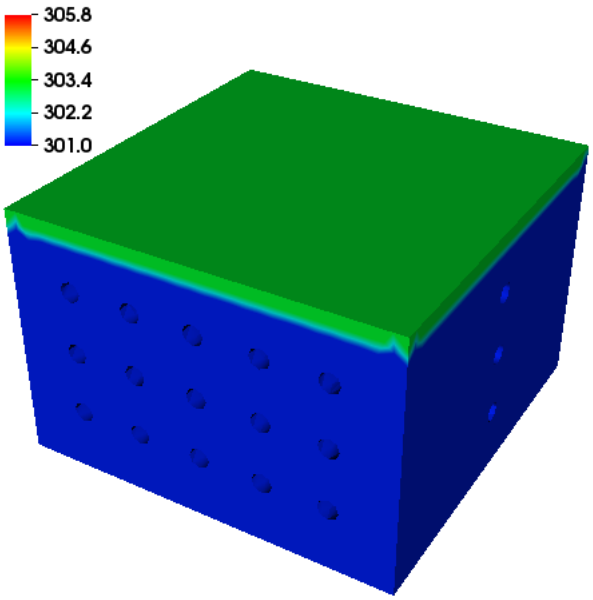


Figure 4.7: Desired state T^* of the homogeneous test scenario.

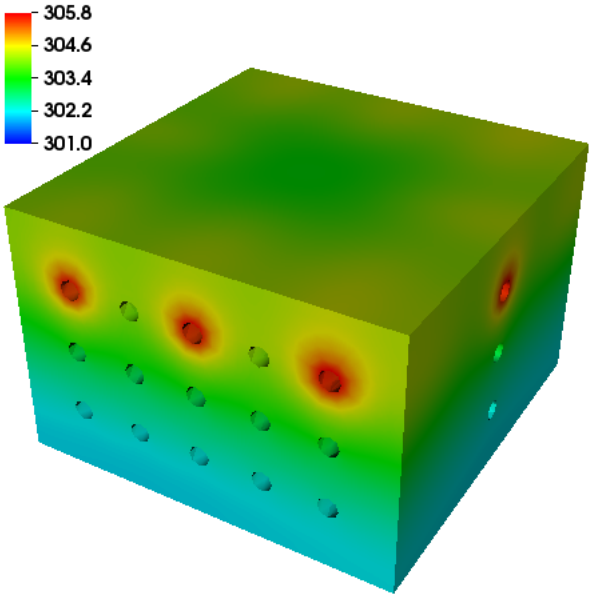


Figure 4.8: Final state $T(t_e)$ of the homogeneous test scenario.

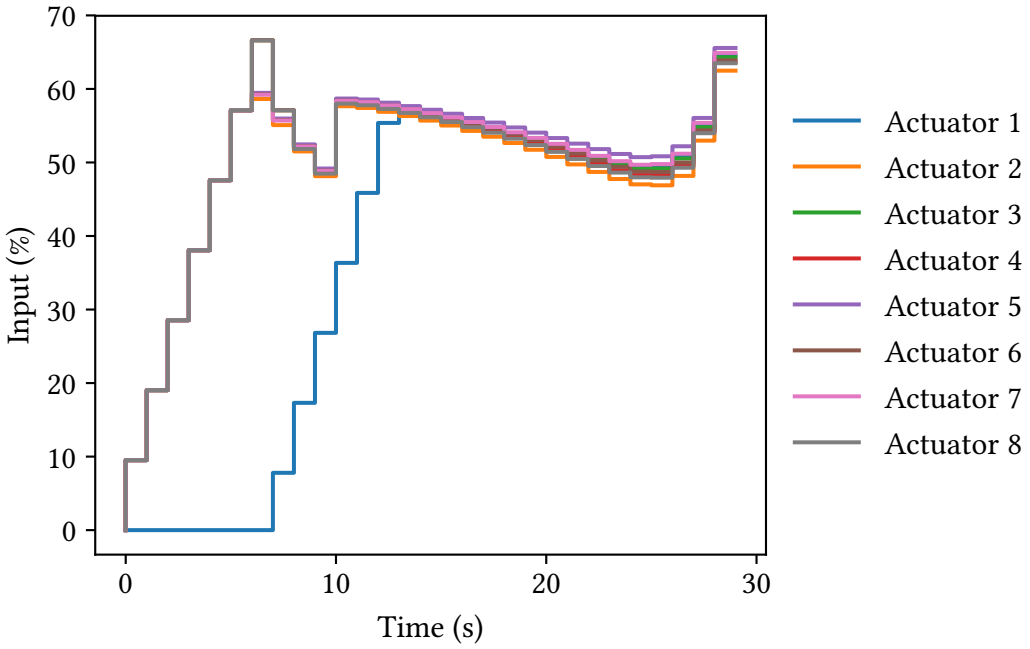


Figure 4.9: Control inputs $\bar{c}$ of the homogeneous test scenario.

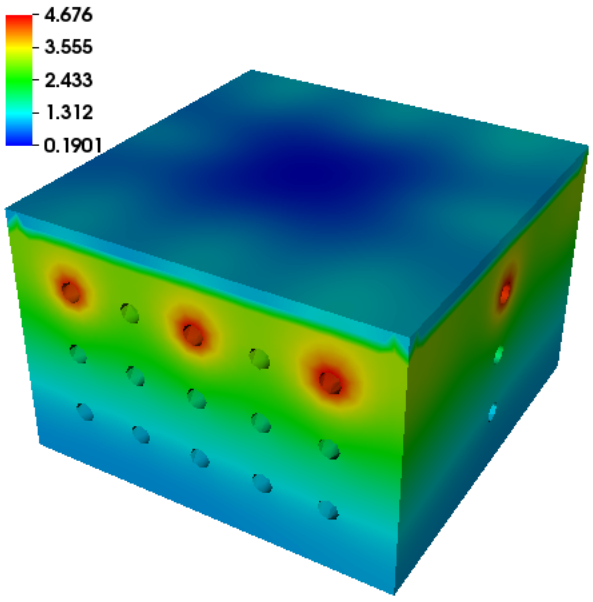


Figure 4.10: Difference of the final state and desired state $T(t_e) - T^*$ of the homogeneous test scenario.

Reviewing the visualizations one can see the chosen desired state in Figure 4.7 which demands a homogeneous temperature at the top of the aluminum block. Since it is clear that this profile is not possible to achieve exactly with the given actuator configuration it just serves as a desired state that can be reached up to a certain error. In Figure 4.8 the simulated solution at the final time $T(t_e)$ using the optimal control input $\bar{c}$ is shown. The computed control inputs $\bar{c}$ for the actuators are depicted in Figure 4.9 and all controls comply with the stated bounds and inequality constraints. All actuators take a very similar trajectory besides actuator 8. Although no real physical reason can be stated for this behavior, since a homogeneous profile is requested, this could be explained by the fact that the achieved profile still overshoots the desired profile by a slight margin and the lagging control compensates for this situation. Note that this is just a possible explanation and no evidence was found. From the visualization it is clear that the top of the block resembles the desired state. Computing the average just on the surface results in

$$\int_{\Gamma_{top}} T(t_e)\, ds = 304\,\mathrm{K}, \tag{4.27}$$

which is an overshoot of 0.85 K and lower than the sensor accuracy. The part below the top surface does not conform with the desired state at all, but this is explained by the cost functional, which is only dependent on the top surface Γ_{top} and does not weigh any contribution of the domain Ω or other parts of the boundary. Figure 4.10 shows the difference of the final state and the desired state $T(t_e) - T^*$. This profile resembles all other observations of the simulation results and one can see the small difference between the achieved and desired state in the considered region. Moving to the experiment, no complete profiles can be shown since no observer technique for reconstruction of the whole state is implemented. Therefore only the sensor measurements are shown in Figure 4.11. It was tried to reflect the circumstances in the simulation as close as possible to the experiment and coincide with environmental temperatures. This is a hard task, because the experiment is not an enclosed space but resides in an open laboratory exposed to non homogeneous conditions regarding air flow. Reviewing Figure 4.11 one can observe the sensor measurements at the top surface recognizing the temperature change from the actuators at about 10 s and reaching an average temperature of about 303.8 K at about 30 s. The results gathered in the physical experiment lead to the conclusion that this approach is successful and is suitable for trajectory planning of heating processes.

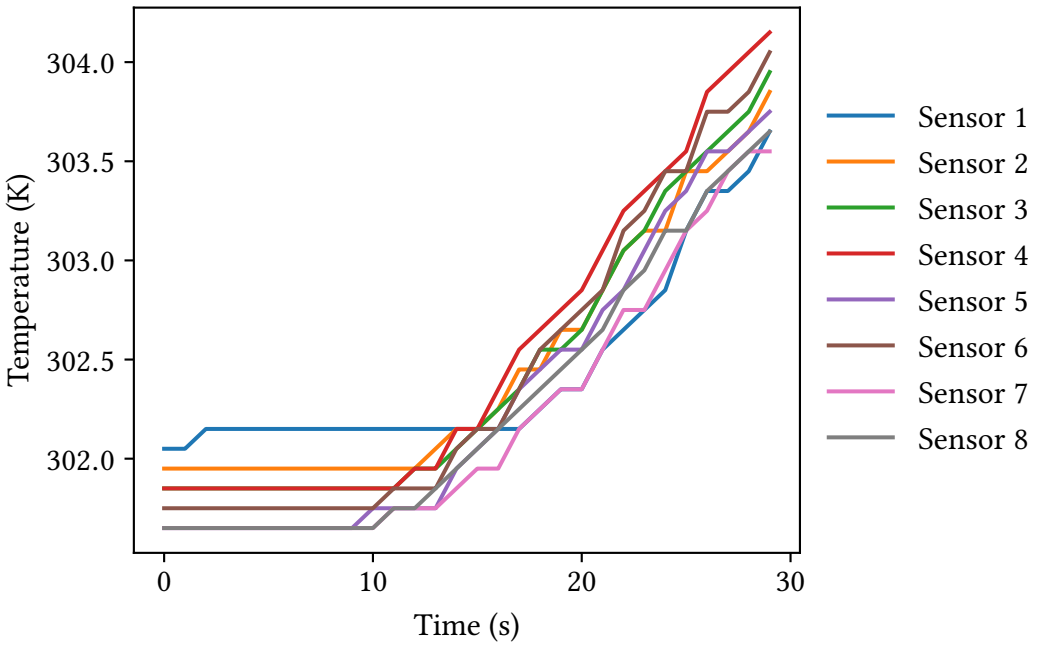


Figure 4.11: Measurements of the sensors on the top surface of the homogeneous test scenario.

4.2.3 Inhomogeneous temperature example

Extending the example described in Section 4.2.2 to form a more complex scenario, an inhomogeneous profile for the top surface is defined as desired state T^*. The optimal control problem is described by using Equation (4.19) and therefore has to comply with the same bounds and constraints. Tolerances and parameters for the optimization algorithm were exactly the same as in the previous example besides the final time which was increased to $t_e = 60\,\text{s}$. To define a smooth transition between the inhomogeneous parts of the profile the smooth step function

$$\Phi(t, a, b) = \Psi(t, a, b)^2(3 - (2\Psi(t, a, b))$$

is used with the saturation function

$$\Psi(t, a, b) = \max(0, \min(1, (t - a)/(b - a))).$$

The parameters are chosen to $a = -0.05$ and $b = 0.075$ to provide a smooth transition from the lower right corner into the remaining part of the profile. This regulates the desired profile T^* and makes it more suitable for the optimization approach than defining discontinuous fields. Therefore the desired profile is defined as

$$T^* = 296.15 + 3\Phi(x, a, b)\Phi(z, a, b).$$

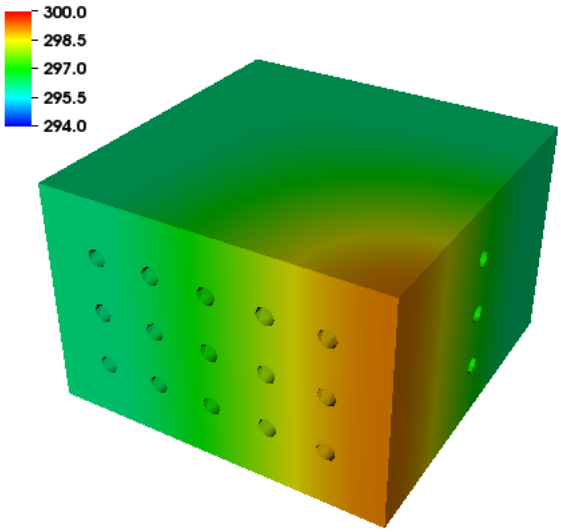


Figure 4.12: Desired state T^* of the inhomogeneous test scenario.

In Figure 4.12 the profile interpolated on the whole block is depicted. Results from the simulation such as the profile at the final time $T(t_e)$ illustrated in Figure 4.13 suggest that the profile is reachable up to a certain error. Looking at Figure 4.14 which shows the difference of the final state and desired state $T(t_e) - T^*$ and the top surface difference in Figure 4.14, it is clear that the error on the top surface is in a reasonable range. Reviewing the control inputs in Figure 4.16 computed by solving the optimal control problem, one can see that the acutators in the lower right corner, namely actuators 3, 4 and 5, receive the highest input percentage to achieve the temperature difference. Other actuators are supporting the temperature increase in other parts of the surface during the trajectory interval. After applying the control inputs to the experiment one can observe the sensor values on the top surface in Figure 4.17. All sensors show the expected behavior and resemble the simulation results. To support this statement, the simulated sensor 6 and real measurement from sensor 6 are depicted in Figure 4.18 along with the sensor uncertainty. The difference in this illustration can be explained by the fact that the real sensor adheres to a PT_1 behavior during the measurement but the simulated sensor just picks the value instantly. Nevertheless the time series of both sensors coincide qualitatively and the trajectory planning approach shows good results even for inhomogeneous profiles.

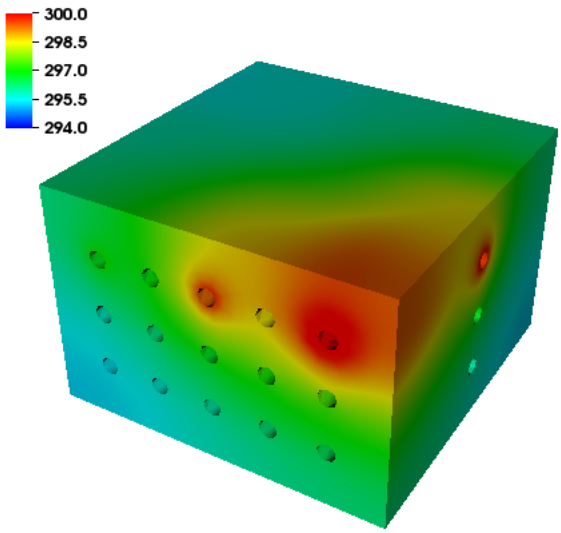


Figure 4.13: Final state $T(t_e)$ of the inhomogeneous test scenario.

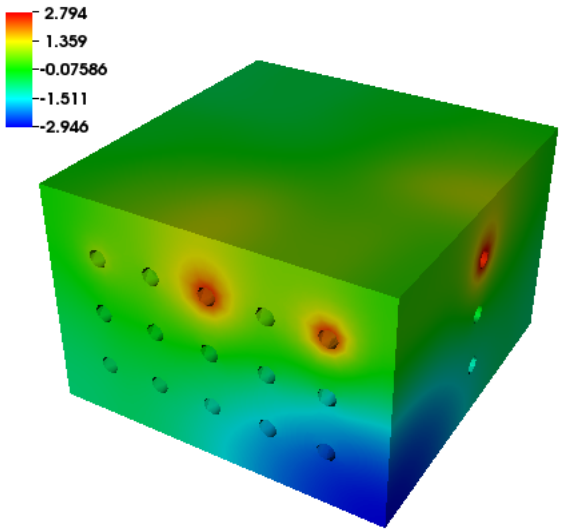


Figure 4.14: Difference of the final state and desired state $T(t_e) - T^*$ of the inhomogeneous test scenario.

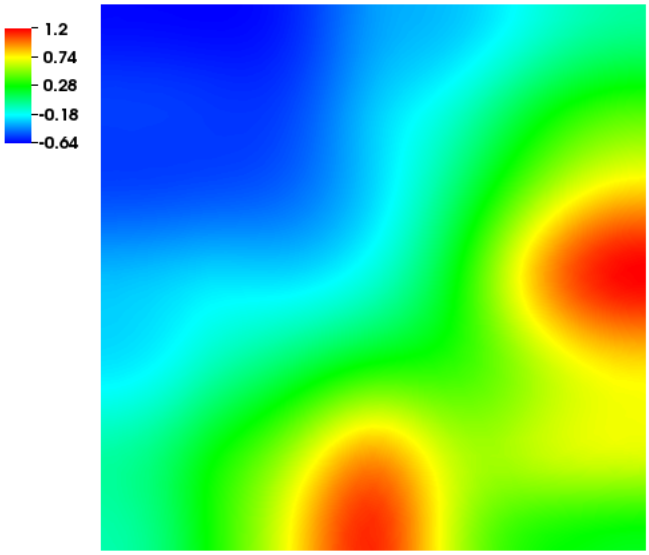


Figure 4.15: Difference of the final state and desired state $T(t_e) - T^*$ at the top surface of the inhomogeneous test scenario.

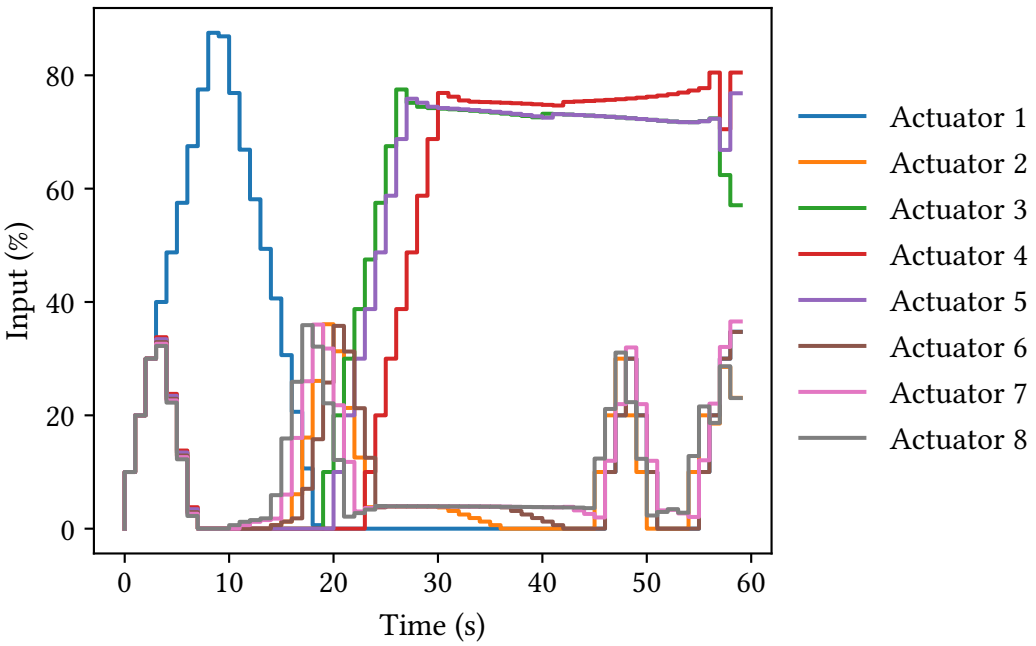


Figure 4.16: Control inputs $\bar{c}$ of the inhomogeneous test scenario.

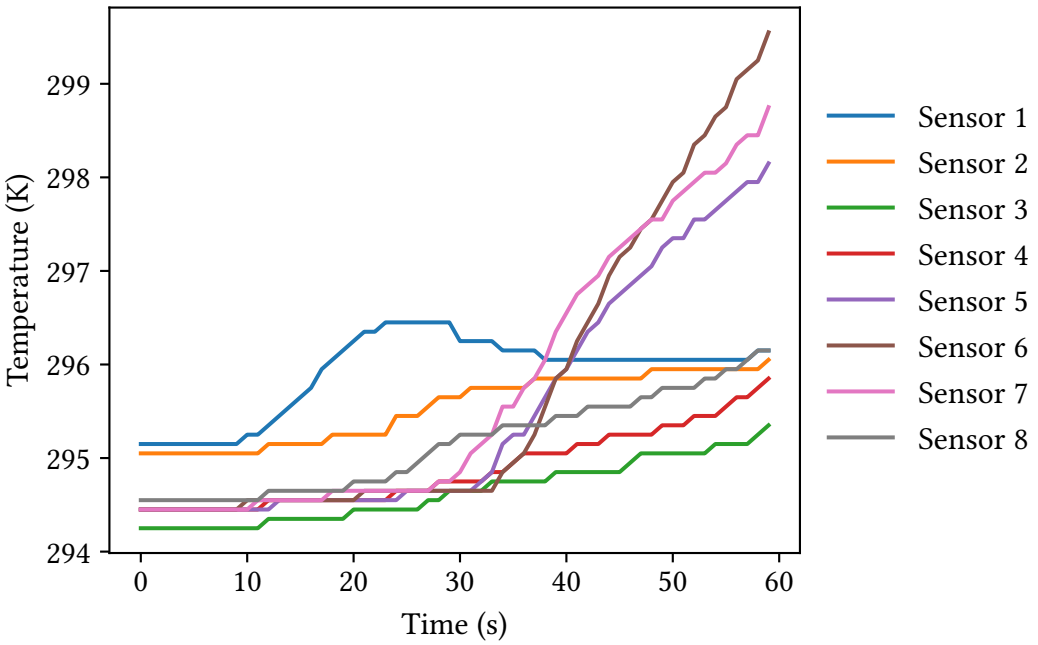


Figure 4.17: Measurements of the sensors on the top surface of the inhomogeneous test scenario.

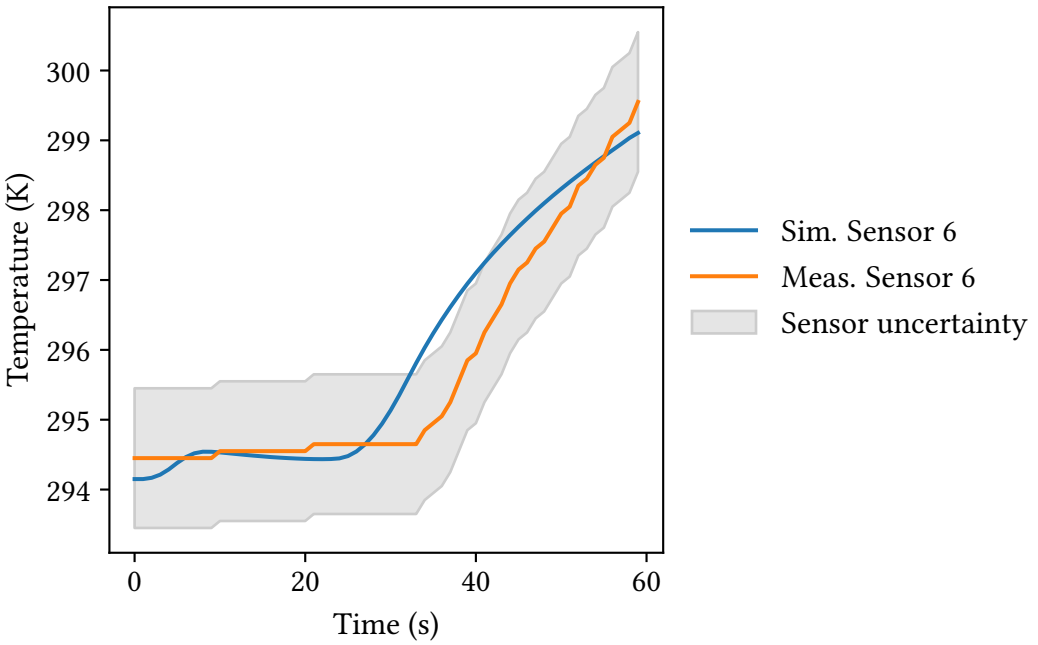


Figure 4.18: Simulation of sensor 6 and measurement data on the top surface of the inhomogeneous test scenario. The uncertainty interval due to the sensor error is depicted as well.

4.3 Air conditioning control in a room

The control of fluid flow is a widely researched topic in the field of PDE constrained optimization (see e.g. [36]). With new methods in discretization for the PDEs and solution techniques for the arising systems, the topic is often revisited. In the context of multiphysics systems we develop an example which involves multiple physical phenomena, namely the fluid flow and temperature in a room as a result of an air conditioning device. We make use of the results from the investigations in Chapter 3 where the influence of loose or tight coupling was shown for the incompressible Navier-Stokes equations and a separate energy equation and apply these to the context of optimization. Due to the forced airflow inlet, the natural convection phenomenon arising from temperature differences in the domain is assumed to be negligable. Therefore the Navier-Stokes equation part of the state equation has no dependency on the temperature and it is feasible to use a loose coupling approach.

Demonstrating this on an example, an optimization problem that has to compute a suitable input trajectory $c(t) \in \mathbb{R}$ for the temperature at the inlet at Γ_s of the air conditioner is defined where a steady inlet velocity is assumed. Consider the input constrained minimization problem for the domain illustrated in Figure 4.19,

$$\min_c J = \int_{\Gamma_s} \frac{\gamma_1}{2} c^2 \, dt + \int_{\Omega} \frac{\gamma_2}{2} (T(t_e) - T^*)^2 \, dx, dt \tag{4.28a}$$

subject to

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \mu \nabla^2 \boldsymbol{u} \qquad \text{in } \Omega \times (t_0, t_e), \tag{4.28b}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \times (t_0, t_e), \tag{4.28c}$$

$$\boldsymbol{u} = \boldsymbol{u}_s \qquad \text{on } \Gamma_s \times (t_0, t_e), \tag{4.28d}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{on } \Gamma_w \times (t_0, t_e), \tag{4.28e}$$

$$\mu \nabla \boldsymbol{u} \cdot \boldsymbol{n} - p\boldsymbol{n} = 0 \qquad \text{on } \Gamma_o \times (t_0, t_e), \tag{4.28f}$$

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \boldsymbol{u} \cdot \nabla T - k \nabla^2 T = 0 \qquad \text{in } \Omega \times (t_0, t_e), \tag{4.28g}$$

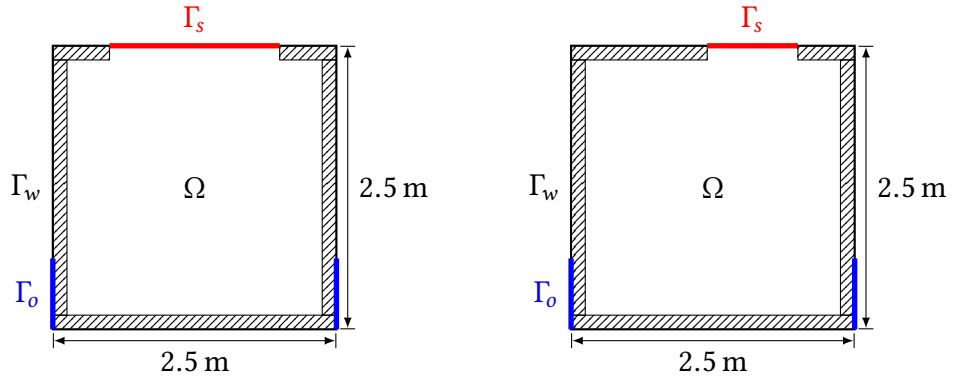
$$\boldsymbol{n} \cdot k \nabla T = h(cQ - T) \qquad \text{on } \Gamma_s \times (t_0, t_e), \tag{4.28h}$$

$$\boldsymbol{n} \cdot k \nabla T = 0 \qquad \text{on } \Gamma_w \cup \Gamma_o \times (t_0, t_e), \tag{4.28i}$$

and

$$-5 \leq \frac{\partial c}{\partial t} \leq 5 \qquad \text{for all } t \in (t_0, t_e), \tag{4.28j}$$

$$T(t_0) \leq c \leq 315.15 \qquad \text{for all } t \in (t_0, t_e). \tag{4.28k}$$



(a) Symmetric case. (b) Asymmetric case.

Figure 4.19: Illustration of the air conditioning control problem in a room with symmetric and asymmetric inlet.

In this problem the control input $c(t)$ is the decision variable which arises as a boundary control term in the energy Equation (4.28h) and is bounded from below by a minimum value $T(t_0)$, which is the initial condition for the temperature for the room, and a maximum value of 315.15 K. Furthermore there is a constraint on the rate of change of 5 K/s to regulate the input and avoid jumps. As one can see in the state equations, there is no direct dependency of the velocity $\boldsymbol{u}$ on the temperature T and therefore the gradient and optimization problem will only be dependent on the temperature due to the control input $c(t)$. We exploit this property by using only the energy equation part in the optimization procedure and precompute the velocity field for the specified time interval. This allows us to adhere to the nonlinear velocity fields arising from the incompressible Navier-Stokes equations but avoid recomputing them in every optimization loop since we assume the influence of the temperature change to be minor to the velocity.

For the optimization process the same approach as in the previous examples is used and consequently an adjoint equation has to be derived in order to compute the gradient. The detailed derivation procedure can be reviewed in Appendix B. The time dependent adjoint equation reads

$$-\rho C_p \frac{\partial \lambda}{\partial t} - \rho C_p \boldsymbol{u} \cdot \nabla \lambda - k \nabla^2 \lambda = 0 \qquad \text{in } \Omega \times (t_0, t_e), \tag{4.29a}$$

$$(\rho C_p \lambda \boldsymbol{u} + k \nabla \lambda) \cdot \boldsymbol{n} + \lambda h = 0 \qquad \text{on } \Gamma_s \times (t_0, t_e), \tag{4.29b}$$

$$(\rho C_p \lambda \boldsymbol{u} + k \nabla \lambda) \cdot \boldsymbol{n} = 0 \qquad \text{on } \Gamma_w \cup \Gamma_o \times (t_0, t_e). \tag{4.29c}$$

The terminal condition is given by

$$\lambda(t_e) = -\frac{\gamma_2}{\rho C_p}(T(t_e) - T^*) \tag{4.30}$$

and the gradient expression reads

$$\frac{d\mathcal{L}}{dc} = \int \gamma_1 c \, dt - \iint_{\Gamma_s} \lambda Q h \, ds dt. \tag{4.31}$$

The forward problem is discretized by using the finite element method discretization presented in Section 3.3 for the Navier-Stokes part of the equation. The energy equation part is discretized as described in Section 2.3. Since both physics are advection dominated they are stabilized using the SUPG method (see Section 1.3) to reduce oscillations in the solution. The same procedure is applied to the adjoint equation. Due to the property of the SUPG method being a consistent stabilization, the solution and thus the gradient is not distorted. This procedure is a key part of the solution process and makes a solution possible.

To make the solution and optimization process more clear we show the necessary steps for the implementation. As mentioned before, the discretized velocity profile $\boldsymbol{u}_h$ is precomputed using the following stabilized variational formulation

Problem 4.2. *Find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ and $p_h \in Q_h$ such that*

$$\mathcal{A}(\boldsymbol{u}_h, \boldsymbol{v}_h, p_h, q_h) + \sum_{K \in \mathcal{T}} \int \tau_0 (\boldsymbol{u}_h \cdot \nabla) \boldsymbol{v}_h \cdot \mathcal{R}_0(\boldsymbol{u}_h, p_h) \, dK = 0 \tag{4.32}$$

for all $\boldsymbol{v}_h \in \boldsymbol{V}_h$ and $q_h \in Q_h$ with $\boldsymbol{V}_h, Q_h$ defined in Equation (3.12) *and $\mathcal{R}_0(\boldsymbol{u}_h, p_h)$ the residual form of the Navier-Stokes equations. The stabilization parameter τ_0 is defined in Equation* (3.19).

We define the function space S_h for the temperature T_h as

$$S_h = \left\{ s_h \in L_2 : s_h|_\tau \in \mathcal{P}_1 \text{ for all } \tau_i \in \mathcal{T}_h \right\}. \tag{4.33}$$

Now we can define the variational problem for the state equation of the advection diffusion part of equation Equation (4.28) with the following problem.

Problem 4.3. *Find $T_h \in S_h$ such that*

$$\begin{aligned} &\left(\rho C_p \frac{\partial T_h}{\partial t}, s_h \right) + (\rho C_p \boldsymbol{u}_h \cdot \nabla T, s) + (k \nabla T, \nabla s) - (h(cQ - T), s)_{\Gamma_s} \\ &+ \sum_{K \in \mathcal{T}} \int \tau_1 (\boldsymbol{u}_h \cdot \nabla) s_h \cdot \mathcal{R}_1(T_h, \boldsymbol{u}_h) \, dK = 0 \end{aligned} \tag{4.34}$$

for all $s_h \in S_h$ and $\mathcal{R}_1(T_h, \boldsymbol{u}_h)$ the residual form of the advection diffusion part of Equation (4.28).

Algorithm 4 Convective heat transfer reduced space optimization

Precompute velocity $\boldsymbol{u}_h \in [t_0, t_e]$ from Problem 4.2
$c(t)^0 \leftarrow$ initial guess
for $k = 1, 2, \ldots$ **do**
  Solve state equation from Problem 4.3 for T
  Solve adjoint equation from Problem 4.4 for λ
  Compute $\frac{d\mathcal{L}}{dc} = \int \gamma_1 c \, dt - \iint_{\Gamma_s} \lambda h \, ds dt$.
  Decide how to change c and continue
end for

The discretized adjoint state $\lambda_h \in S_h$ is computed by the following variational problem.

Problem 4.4. *Find $\lambda_h \in S_h$ such that*

$$
\begin{aligned}
&- \left(\rho C_p \frac{\partial \lambda_h}{\partial t}, s_h \right) + (\rho C_p \boldsymbol{u}_h \cdot \nabla s, \lambda) + (k \nabla T, \nabla s) + (h\lambda, s)_{\Gamma_s} \\
&+ \sum_{K \in \mathcal{T}} \int \tau_1 (\boldsymbol{u}_h \cdot \nabla) s_h \cdot \mathcal{R}_2(\lambda_h, \boldsymbol{u}_h) \, dK = 0
\end{aligned} \tag{4.35}
$$

for all $s_h \in S_h$ and $\mathcal{R}_2(\lambda_h, \boldsymbol{u}_h)$ the residual form of Equation (4.29).

Note that in the derivation of the adjoint state, the advection term is integrated by parts because of the arising boundary condition and therefore different from the forward state variational form. The stabilization parameter τ_1 is defined by

$$
\tau_1 = \left(\frac{4k}{h^2} + \frac{2|\boldsymbol{u}_h|}{h} \right)^{-1}, \tag{4.36}
$$

which has shown to provide decent stability in this scenario. Finally, all parts of the equations are discretized in time by BDF-1 with equidistant time steps on the same grid for the given time interval. The implementation is done in the `Firedrake` framework. The optimization procedure is illustrated in Algorithm 4. Since the `SNOPT` library is used for optimization, this does not exactly resemble how often the forward and adjoint equations and therefore the gradient is evaluated. Furthermore it has to be emphasized that Problems 4.2 to 4.4 are solved using a parallel implementation. The results for the serial optimization algorithm implemented in `SNOPT` are collected on the root process and evaluated on that node. This allows to analyze large scale systems, as present in this case, in a reasonable time frame.

4.3.1 Symmetric inlet

In the first example we consider a symmetric velocity and temperature inlet Γ_s for a two dimensional slice of a room with $2.5\,\mathrm{m} \times 2.5\,\mathrm{m}$ shown in Figure 4.19a. Velocity outlets are defined in the left and right bottom corner on the boundaries Γ_o. For the parameters of the fluid we consider air with density $\rho = 1.244\,\mathrm{kg/m^3}$, kinematic viscosity $\mu = 1.8 \times 10^{-5}\,\mathrm{kg/m\,s}$, specific heat capacity $C_p = 1005.0\,\mathrm{J/kg\,K}$ and thermal conductivity $0.0253\,\mathrm{W/m\,K}$. The velocity at the inlet is fixed at $\boldsymbol{u}_s = -0.4\,\mathrm{m/s}$ and the initial condition is chosen as $\boldsymbol{u}(t_0) = \mathbf{0}$. The initial temperature of the room is $T(t_0) = 291.15\,\mathrm{K}$ and the desired temperature after a time frame of 15 s is set to $T^* = 296.15\,\mathrm{K}$. As in Section 4.2.2, the optimization algorithm of `SNOPT` is used with the same tolerance of 1×10^{-4}. The regularization parameters in the objective functional are set to $\gamma_1 = 1 \times 10^{-6}$ and $\gamma_2 = 1 \times 10^{2}$. In this test case the number of actuators is $N_a = 1$ and with an equidistant time step of $\Delta t = 0.05$ the number of decision variables for the optimization problem is

$$N_a \times N_t = 300.$$

In Figure 4.20 the computed control input $c(t_k)$ is shown along with the resulting temperature at the inlet $T|_{\Gamma_s}$ and the average temperature in the whole domain $\bar{T}|_{\Omega}$. The resulting temperature profiles are shown in Figure 4.21 for different time steps. In this idealized case of a large inlet for velocity and temperature compared to the whole size of the domain, one can observe that the computed control can achieve the desired state up to a small difference.

4.3.2 Asymmetric inlet

The second example, illustrated in Figure 4.19b, only differs from the prior by narrowing the inlet boundary Γ_s and consequently providing an asymmetric inflow of velocity and temperature and solves the optimization problem in Equation (4.28). In Figure 4.22 the computed control input $c(t_k)$ is depicted along with the resulting temperature at the inlet $T|_{\Gamma_s}$ and the average temperature in the whole domain $\bar{T}|_{\Omega}$. The control input adheres to the bounds and constraints prescribed by the optimization problem. Looking at the average temperature in Figure 4.22 the desired temperature of $T^* = 296.15\,\mathrm{K}$ is not reached completely after $t = 15\,\mathrm{s}$ but the acquired solution lies within a feasible region. In Figure 4.23 the temperature profiles at different time steps are shown resulting from the computed control input. The transient temperature profile behavior emphasizes the fact that the asymmetric inlet can not reach the desired temperature in the same time frame as the larger symmetric inlet. To reduce the difference even further, one could increase the time frame for the optimization problem or increase the inlet velocity.

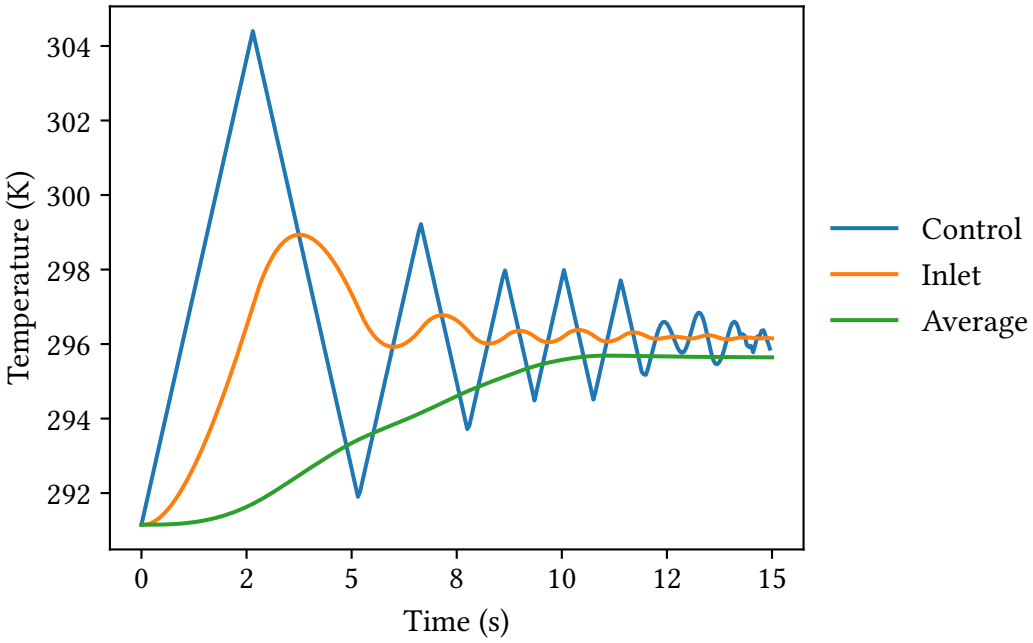
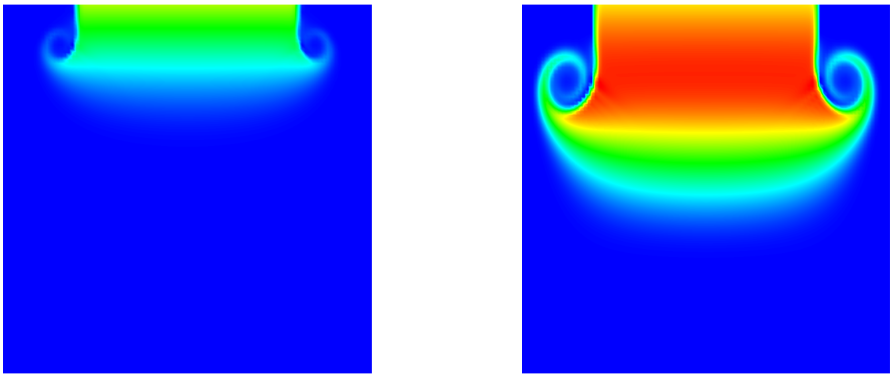
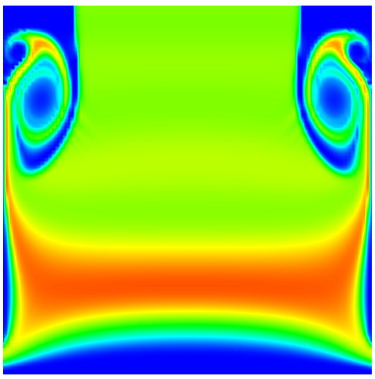


Figure 4.20: Control input $c(t_k)$, Inlet temperature $T|_{\Gamma_s}$ and average temperature $\bar{T}|_\Omega$ in the symmetric case.

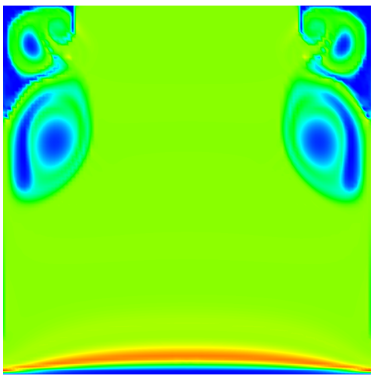


(a) Temperature at $t = 2.5$ s.

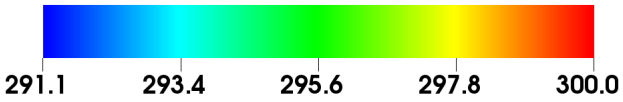
(b) Temperature at $t = 5$ s.



(c) Temperature at $t = 10$ s.



(d) Temperature at $t = 15$ s.



(e) Color bar for the temperature profile range.

Figure 4.21: Temperature profiles resulting from the computed control applied to the air conditioning problem with an symmetric inlet at different time steps.

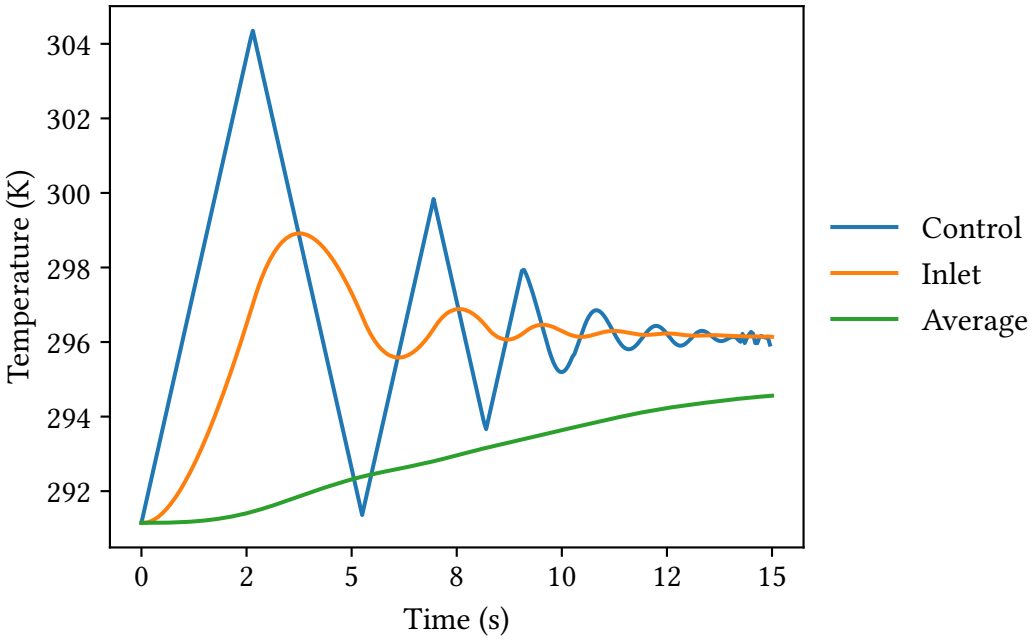
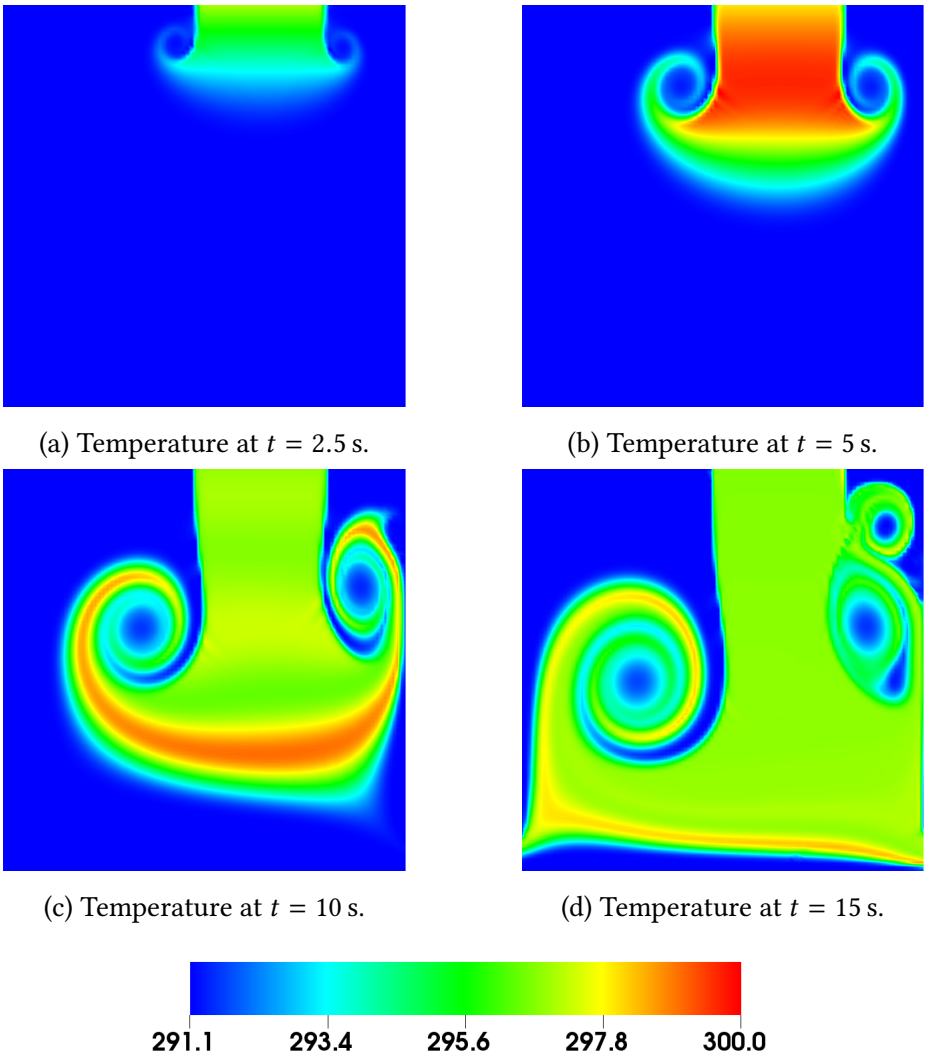


Figure 4.22: Control input $c(t_k)$, Inlet temperature $T|_{\Gamma_s}$ and average temperature $\bar{T}|_\Omega$ in the asymmetric case.



(a) Temperature at $t = 2.5$ s.

(b) Temperature at $t = 5$ s.

(c) Temperature at $t = 10$ s.

(d) Temperature at $t = 15$ s.

(e) Color bar for the Temperature profile range.

Figure 4.23: Temperature profiles resulting from the computed control applied to the air conditioning problem with an asymmetric inlet at different time steps.

Chapter 5

Flatness-based optimal control

The techniques derived in Chapter 4 are suitable for trajectory planning or feedback control techniques like model predictive control [35]. An alternative strategy to solve a trajectory planning problem is to exploit the concept of differential flatness [30]. This technique is based on theory for finite dimensional systems but has been extended for use with partial differential equations in recent years including linear and nonlinear diffusion-reaction systems [57, 54, 75] or hyperbolic systems [64, 87] and flexible structures [76, 58]. Most examples are shown for one dimensional systems, although there are references that treat higher dimensional domains [56, 10, 58]. Parametrization of the state has been studied in a variety of publications and there exist several techniques depending on the equation properties. For linear systems the Laplace transformation and spectral methods [68, 20] along with eigenvalue decompositions [58] have been used. Nonlinear systems have been treated using formal power series [24, 61] or formal integration [56] approaches. Reviewing most of the references suggest that a Gevrey function is chosen for the flat output due to the requirements imposed by the transition restrictions. In the general flatness-based design of trajectory planning algorithms one cannot incorporate constraints directly. To address optimization constraints on the trajectory or the state the authors in [22, 79] use splines for parametrization of the trajectory and a nonlinear model predictive control approach is implemented for constraint handling. Similar approaches to circumvent the restrictions of Gevrey functions consist of spectral methods using Legendre polynomials [70] with Gauss-Legendre-Lobatto nodes which in combination yield high order accuracy for derivatives.

5.1 Constrained trajectory planning

An alternative to the mentioned approaches from [22, 79, 70], where splines or spectral methods are used, is the approach of [49]. On the example of a finite dimensional system, the authors utilize piecewise constant functions to express the highest derivative of the flat output and formulate an integrator chain system to describe the derivatives. In this work we extend the method of [49] to an example including partial differential equations, a diffusion-reaction system in particular. The formulation using an integrator chain is appealing because it allows the incorporation of state and input constraints, that have to be satisfied by the computed trajectory, by design. All derivations and results were published by the author in [4].

5.1.1 Problem formulation

As an illustrative example consider the linear diffusion-reaction system

$$\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2} + rq \qquad \text{for } x \in (0,1)\,, t > 0, \tag{5.1a}$$

$$\frac{\partial q}{\partial x} = 0 \qquad \text{on } x = 0, \tag{5.1b}$$

$$q = u \qquad \text{on } x = 1, \tag{5.1c}$$

$$q(t_0) = q^0, \qquad \text{in } x \in [0,1] \tag{5.1d}$$

that describes the quantity $q(x,t)$ on the one-dimensional domain $x \in [0,1]$. The constant scalar parameter $r \in \mathbb{R}$ can be set to 0 and Equation (5.1) results in the linear heat equation with all material depending parameters set to 1. This can be achieved, without loss of generality, by properly scaling x, t and q. The notation here is used on purpose since we only deal with one dimensional problems in this chapter. The purpose of planning a trajectory is to transfer the state $q(t_0)$ to a desired state $q^*(t_e)$ in a finite time t_e using a precomputed input $u(t)$. The end time does not have to be fixed and is determined by the optimization process in this approach. We can formulate an optimal control problem with the objective functional

$$\min_u J = \int_0^{t_e} \int_0^1 l(t,q,u)\, dx dt \tag{5.2a}$$

subject to (5.1) and the state and input constraints

$$q^- \leq q \leq q^+ \quad \text{for all } t \geq 0, \quad x \in [0,1] \tag{5.2b}$$

$$u^- \leq u \leq u^+ \quad \text{for all } t \geq 0. \tag{5.2c}$$

As one can see this is a very similar optimal control problem to Equation (4.19). Finding a solution to Equation (5.2a) results in a trajectory $\bar{u}$ that provides an open-loop input for Equation (5.1). In this work the method of [49] is extended to be applicable to Equation (5.2a) that involves a partial differential equation as equality constraint. Therefore we employ a flatness-based formulation of Equation (5.1) by using the definition from [30].

Definition 5.1 (Differential Flatness)**.** The nonlinear system with $z \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$

$$\dot{z}(t) = f(z(t), u(t)), \quad t > 0 \tag{5.3}$$
$$z(0) = z^0 \in \mathbb{R}^n \tag{5.4}$$

is called flat, if there exists a flat, also called basic output y with $\dim y = \dim u = m$ enabling a differential parametrization of the states $z(t)$ and the inputs $u(t)$ in the form

$$z(t) = \theta_z \left(y(t), y^{(1)}(t), \ldots, y^{(\beta-1)}(t) \right), \tag{5.5}$$
$$u(t) = \theta_u \left(y(t), y^{(1)}(t), \ldots, y^{(\beta)}(t) \right) \tag{5.6}$$

for some tuple $\beta = (\beta_1, \ldots, \beta_m)$ with $\sum_{j=1}^{m} \beta_j \geq \dim z$.

We can also employ the techniques from [51, 57] to derive a flat output for the diffusion-reaction system Equation (5.1) based on the formal power series

$$q(x, t) = \sum_{n}^{\infty} a_n(t) \frac{x^n}{n!}. \tag{5.7}$$

We substitute Equation (5.7) in Equation (5.1), evaluate the resulting term and sort terms of equal degree in x. This yields the recursion formula

$$a_{n+2}(t) = \dot{a}_n(t) - r a_n(t), \quad n \geq 0.$$

Furthermore the boundary conditions have to be fulfilled and the series coefficient corresponding to Equation (5.1b) is $a_1(t) = 0$. The coefficient $a_0(t)$ is set to $a_0(t) = y(t)$ which is some function that should serve as a flat output later on. This recursion admits a closed-form solution

$$a_{2n}(t) = \sum_{j=0}^{n} \binom{n}{j} (-r)^{n-j} y^{(j)}(t), \quad a_{2n+1}(t) = 0.$$

Substituting the derived coefficients in Equation (5.7) yields a state and input parametrization

$$q(x,t) = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} \sum_{j=0}^{n} \binom{n}{j} (-r)^{n-j} y^{(j)}(t), \tag{5.8}$$

$$u(t) = q(1,t), \tag{5.9}$$

which is described by the flat output $y(t)$ due to Definition 5.1. The general approach in [30] and the references therein would be to find a suitable function for $y(t)$, that has very strict requirements in most cases. In particular $y(t)$ has to be infinitely differentiable. A popular choice is a Gevrey class function (see e.g. [59]). A particular example is the function $\Phi(t)$ that is locally non-analytic for $t \in [0,t]$

$$\Phi(t) = \begin{cases} 0 & t \leq 0, \\ \frac{\int_0^t h_{T,\omega}(p)dp}{\int_0^T h(p)dp} & t \in (0,T), \\ 1 & t \geq T, \end{cases} \tag{5.10}$$

and

$$h(t) := \exp\left(-\left(\frac{\left(1-\frac{t}{T}\right)t}{T}\right)^{\omega}\right). \tag{5.11}$$

This function can only be modified with the parameters T, that basically resembles the end time of the trajectory, and the slope ω. Therefore the possibilities to fulfill more criteria than the transition between two known steady states is very limited.

To incorporate the state and input constraints defined in Equations (5.2b) and (5.2c) we use a piecewise constant function to express the highest derivative of the flat output $y(t)$. In the following we also omit the domain and time dependency on known equations for readability. Since the piecewise constant function has to be at least β-times differentiable, an integrator chain formulation is imposed by

$$\begin{aligned} \dot{z} &= \underbrace{\begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{A} z + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{b} v \\ z(0) &= z^0 \in \mathbb{R}^{\beta} \end{aligned} \tag{5.12a}$$

with the state

$$z = \left[y, y^{(1)}, \dots, y^{(\beta-2)}, y^{(\beta-1)}\right]^T \tag{5.12b}$$

and input

$$v = y^{(\beta)}. \tag{5.12c}$$

The here defined function v is the highest derivative and the system described in Equation (5.12a) makes it possible to enforce restrictions on y and its derivatives when incorporated in an optimization procedure. This makes it possible to formulate equality constraints that ensure a transition form the steady state $z(0)$ to the steady state $z(t_e)$ by setting all derivates of y to zero at $t = 0$ and $t = t_e$ with

$$z(0) = \begin{bmatrix} y(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad z(t_e) = \begin{bmatrix} y(t_e) \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{5.13}$$

We can now use the parametrization of q and u with y with a finite number of derivatives (which approximates the infinite sum) and replace the state z with the augmented state $\hat{z} = [z^T, v]^T$ so that

$$q(\hat{z}) = \sum_{n=0}^{\beta} \frac{x^{2n}}{(2n)!} \sum_{j=0}^{n} \binom{n}{j} (-r)^{n-j} \hat{z}_{j+1}(t), \tag{5.14}$$

$$u(\hat{z}) = q(\hat{z}) \quad \text{on } x = 1. \tag{5.15}$$

The optimal control problem Equation (5.2a) has to be reformulated using the augmented state $\hat{z}$ which results in

$$\min_{v} J = \int_0^{t_e} \int_0^1 l(t, q(\hat{z}), u(\hat{z}))\, dx dt \tag{5.16a}$$

subject to Equation (5.12a), i.e.

$$\dot{z} = Az + bv \tag{5.16b}$$

and state and input constraints

$$q^- \leq q(\hat{z}) \leq q^+ \quad \text{for all } t \geq 0, \quad x \in [0, 1] \tag{5.16c}$$

$$u^- \leq u(\hat{z}) \leq u^+ \quad \text{for all } t \geq 0. \tag{5.16d}$$

as well as initial and terminal values in (5.13).

Remark 10. The solution of Equation (5.16a) does not involve the computation of a solution of the partial differential equation given in Equation (5.1). The procedure is rather a shaping of an input trajectory parameterized by $\hat{z}$ that fulfills all conditions of a flat output.

Remark 11. In some applications it is desirable to not specify an end time t_e especially in situations where one is not able to predict an accurate time interval that could lead to a feasible trajectory. Therefore the mapping

$$t \mapsto \tau(t) = \frac{t}{t_e}. \tag{5.17}$$

is introduced to normalize the time variable to the unit time interval. This mapping is applied to the time variable t in Equation (5.16) resulting in

$$\min_{v(\tau),\, t_e} J = \int_0^1 \int_0^1 l(\tau, q(\tau; \hat{z}(\tau)), u(\tau; \hat{z}(\tau)))dxd\tau \tag{5.18a}$$

with the augmented state $\hat{z}(\tau) = [z^T(\tau), v(\tau)]^T$ subject to

$$\frac{\partial z(\tau)}{\partial \tau} = \tilde{A}z(\tau) + \tilde{b}v(\tau), \quad \tau > 0 \tag{5.18b}$$

$$z(0) = z^0 \tag{5.18c}$$

with $\tilde{A} = t_e A$, $\tilde{b} = t_e b$ and the state and input constraints

$$q^- \leq q(\tau; \hat{z}(\tau)) \leq q^+ \quad \tau \in [0,1], \quad x \in [0,1], \tag{5.18d}$$

$$u^- \leq u(\tau; \hat{z}(\tau)) \leq u^+ \quad \tau \in [0,1] \tag{5.18e}$$

and initial as well as terminal values according to Equation (5.13).

This formulation introduces t_e as another independent variable in Equation (5.18a) next to the flat outputs highest derivative normalized in time $v(\tau)$.

5.1.2 Discretization

For the solution of Equation (5.18) numerical methods for optimizations are used. Therefore the problem has to be discretized in time and the solution is described as piecewise constant with the sample time T_s. Let $z_k = z(kT_s)$, $v_k = v(kT_s)$ be the discrete time sampled solution and input of the integrator chain system we can describe

$$z_{k+1} = \Phi z_k + \boldsymbol{\gamma} v_k, \quad k \geq 0 \tag{5.19a}$$

$$z_0 = z^0 \tag{5.19b}$$

where

$$\Phi = \exp(\tilde{A}T_s) = \begin{bmatrix} 1 & t_e T_s & \dots & \frac{(t_e T_s)^{\beta}}{\beta!} \\ 0 & 1 & \dots & \frac{(t_e T_s)^{\beta-1}}{(\beta-1)!} \\ \vdots & & & \vdots \\ 0 & 0 & \dots & t_e T_s \\ 0 & 0 & \dots & 1 \end{bmatrix}, \tag{5.19c}$$

and

$$\begin{aligned} \gamma &= \int_{kT_s}^{(k+1)T_s} \exp\left(\tilde{A}[(k+1)T_s - s]\right)\tilde{b}ds \\ &= \left[\frac{(t_e T_s)^{\beta+1}}{(\beta+1)!} \quad \frac{(t_e T_s)^{\beta}}{\beta!} \quad \dots \quad t_e T_s\right]^T. \end{aligned} \tag{5.19d}$$

Introducing the same sampling variables in the objective functional in Equation (5.18a) and replacing the time integral with an appropriate approximation using a sum representation with weight a_k, using the Simpson or trapezoidal rule, this yields

$$J = \int_0^1 \sum_{k=0}^{N} a_k l(kT_s, q_k(\hat{z}_k), u_k(\hat{z}_k))dx, \tag{5.20}$$

with $k = 0, 1, \dots, N$ and $N = 1/T_s$. Finally we have to define the discretized parametrizations of the state and input equation with

$$q_k(\hat{z}_k) = \sum_{n=0}^{\beta} \frac{x^{2n}}{(2n)!} \sum_{j=0}^{n} \binom{n}{j} (-r)^{n-j} \hat{z}_{k,j+1} \tag{5.21}$$

$$u_k(\hat{z}_k) = q_k(\hat{z}_k) \quad \text{on } x = 1 \tag{5.22}$$

with $\hat{z}_{k,j+1}$ the j-th element of the vector $\hat{z}_k$. In order to clarify clear how Equation (5.19) is evaluated we list the state variable at each k for $0, 1, \ldots, N$

$$\begin{aligned} z_0 &= {}^0 \\ z_1 &= \Phi z^0 + \gamma v_0 \\ z_2 &= \Phi^2 z^0 + \begin{bmatrix} \Phi\gamma & \gamma \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} \\ &\vdots \\ z_N &= \Phi^N z^0 + \begin{bmatrix} \Phi^{N-1}\gamma & \ldots & \Phi\gamma & \gamma \end{bmatrix} \begin{bmatrix} v_0 \\ \vdots \\ v_{N-1} \end{bmatrix}. \end{aligned} \tag{5.23}$$

Next, the matrix

$$S_k = \begin{bmatrix} \Phi^{k-1}\gamma & \ldots & \Phi\gamma & \gamma \end{bmatrix} \in \mathbb{R}^{\beta \times k}$$

is introduced. Using the vector

$$V_k = \begin{bmatrix} v_0 & v_1 & \ldots & v_{k-1} \end{bmatrix}^T \in \mathbb{R}^k$$

we obtain

$$\hat{z}_k = \begin{bmatrix} z_k \\ v_k \end{bmatrix} = \begin{bmatrix} S_k \\ e_k^T \end{bmatrix} V_k + \begin{bmatrix} \Phi^k \\ 0^T \end{bmatrix} z^0, \tag{5.24}$$

where $e_k^T = [0, \ldots, 0, 1] \in \mathbb{R}^{1 \times k}$. Reformulating the integrator chain system in this way highlights the only unknown $\hat{z}(V_k)$ in the formulation which is the full vector of unknown constant coefficients yielding the discretized highest derivative of the flat output. Finally the fully time discretized optimization problem can be denoted by replacing $\hat{z}_k$ in Equation (5.20) with Equation (5.24)

$$\min_{V_k,\, t_e} J = \sum_{k=0}^{N} a_k l(kT_s, q_k(V_k), u_k(V_k)), \tag{5.25a}$$

subject to state and input constraints

$$q^- \leq q_k(V_k) \leq q^+ \quad x \in [0, 1], \tag{5.25b}$$

$$u^- \leq u_k(V_k) \leq u^+, \tag{5.25c}$$

for $k = 0, 1, \ldots, N$ and the terminal condition

$$z_N = \begin{bmatrix} I & 0 \end{bmatrix} \hat{z}_N(V_N) = z(t_e), \tag{5.25d}$$

where I is the $(\beta \times \beta)$ identity matrix. The optimization problem in this form yields a dimension of $\dim V_k + \dim t_e = N + 1$ independent variables.

Scenario	T_s	r	β	α	u^+	t_e^+
1	0.0125	0.0	5	1×10^{-8}	1.2	2.0
2	0.0125	0.0	5	1×10^{-8}	1.2	1.35
3	0.0125	-2.0	5	1×10^{-8}	0.4	4.0
4	0.0125	0.0	5	1×10^{-20}	1.2	2.0
4	0.0125	0.0	5	1×10^{-10}	1.2	2.0
4	0.0125	0.0	5	1×10^{-8}	1.2	2.0
4	0.0125	0.0	5	1.0	1.2	2.0

Table 5.1: Simulation parameters for the flatness-based optimal control scenarios.

5.1.3 Examples

Consider the reaction-diffusion system in Equation (5.1) and the derivation of a flat output parameterized by the integrator chain method described in the previous chapter. In the following examples Equation (5.1) is discretized using the `Firedrake` framework to obtain a simulation model to test the computed trajectories. The arising optimization problem is solved using `IPOPT` which implements a primal-dual interior point method [84].

The optimal control problem is described by

$$\min J = \frac{1}{2}\left(\alpha \sum_{k=0}^{N-1} v_k^2 + t_e^2\right) \tag{5.26a}$$

subject to

$$q^- \leq q_k \leq q^+ \quad \forall k \in [0, \ldots, N], \tag{5.26b}$$
$$0 \leq u_k \leq u^+ \quad \forall k \in [0, \ldots, N], \tag{5.26c}$$
$$0 < t_e \leq t_e^+ \tag{5.26d}$$
$$y_0^{(n)} = 0 \quad \forall n \in [1, \ldots, \beta], \tag{5.26e}$$
$$y_N^{(n)} = 0 \quad \forall n \in [1, \ldots, \beta]. \tag{5.26f}$$

Reviewing Equation (5.26), the end time t_e is not fixed and is determined during the optimization routine and bounded by Equation (5.26d). The primary independent variables v_k are regulated using a Tikhonov parameter α, which serves the purpose of scaling the objective functional to cope with possible numerical difficulties in the optimization procedure.

The simulation parameters can be obtained from Table 5.1 and for every trajectory planning problem we consider the initial state $q(x, \tau = 0) = 0$ and

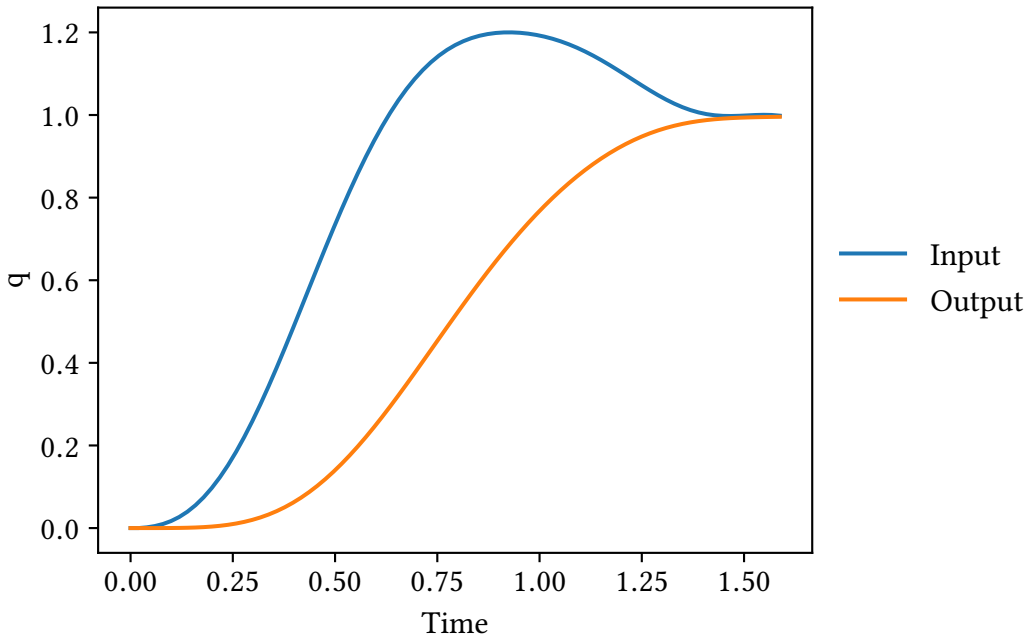


Figure 5.1: Input and output trajectory of scenario 1.

define a desired final state $q^*(0, t_e) = 1$. The sampling time is set to $T_s = 0.0125$ so that $N = 80$. A crucial point in the approximation is that we require the flat output y to be at least $\beta = 5$-times differentiable. This reveals a trade-off in the choice of time domain discretization points and differentiability. In the test scenarios the Thikonov regularization parameter is always set to $\alpha = 1 \times 10^{-8}$.

In the first scenario the reaction parameter is set to $r = 0$ and therefore the dynamics resemble pure diffusion. The upper bound of the input is set to $u^+ = 1.2$ and the final time is restricted to $t_e^+ = 2.0$. Reviewing the depicted results of the control input and corresponding output in Figure 5.1 show that a reasonable input can be computed by the method that leads to the desired output $q^*(0, t_e)$ in a time frame that fulfills $t_e \leq t_e^+$. A space and time solution profile is shown in Figure 5.2.

Since the control input does not seem to exhaust the bounds and constraints, a second scenario that restricts the upper bound of the final time even further with $t_e^+ = 1.35$ is employed. The results are shown in Figures 5.3 and 5.4. This makes clear that this restriction is demanding on the control input and the value is even touching the bounds so that no further decrease in the upper bound is considered.

In the last scenario we show the behavior of a reaction parameter $r = -2.0$. Herein the control input upper bound is reduced significantly to $u^+ = 0.4$, since we assume the reaction is generating energy in the system and assist the control.

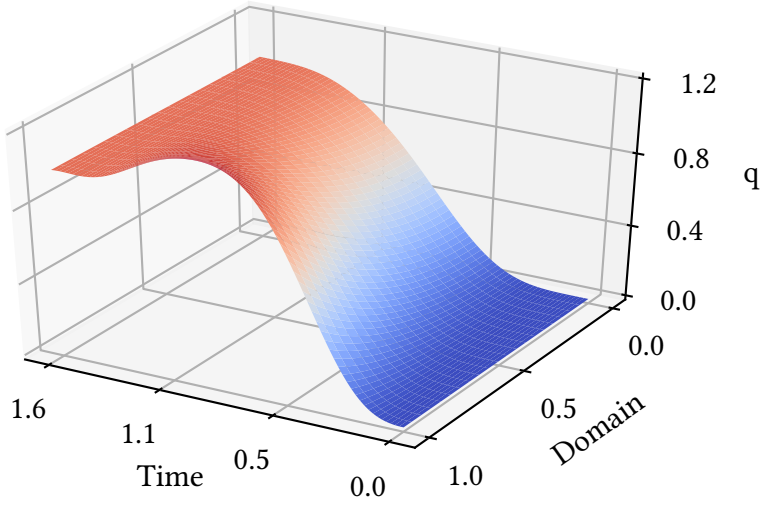


Figure 5.2: Profile of the solution of scenario 1.

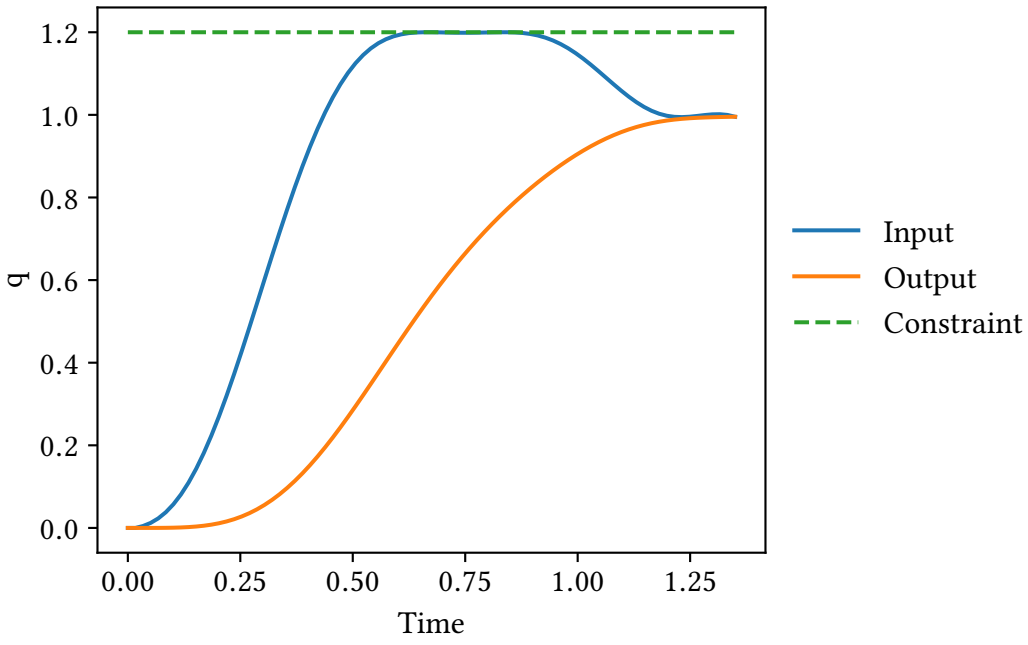


Figure 5.3: Input and output trajectory of scenario 2.

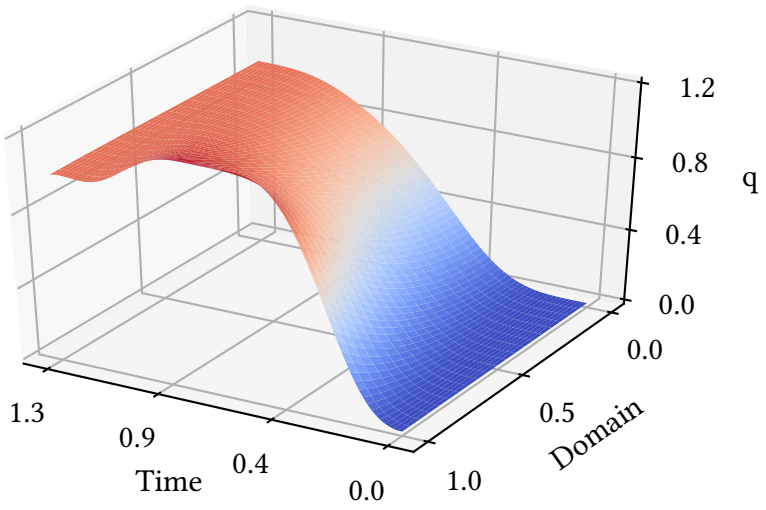


Figure 5.4: Profile of the solution of scenario 2.

Furthermore the upper bound on the final time is increased to $t_e^+ = 4.0$ to compensate the reduced control input. As one can observe in Figures 5.5 and 5.6 the method is able to generate a control input trajectory that fulfills constraints in the control input and state to meet the trajectory planning criteria. To demonstrate the impact of the Thikhonov regularization parameter α in the objective functional the first scenario is repeated using the different values from Table 5.1 corresponding to scenario 4. The results can be observed in Figures 5.7 and 5.8. Analyzing the behavior it shows a decreasing penalization of the input trajectory and the weighting of the final time t_e^+ increases with shrinking α values.

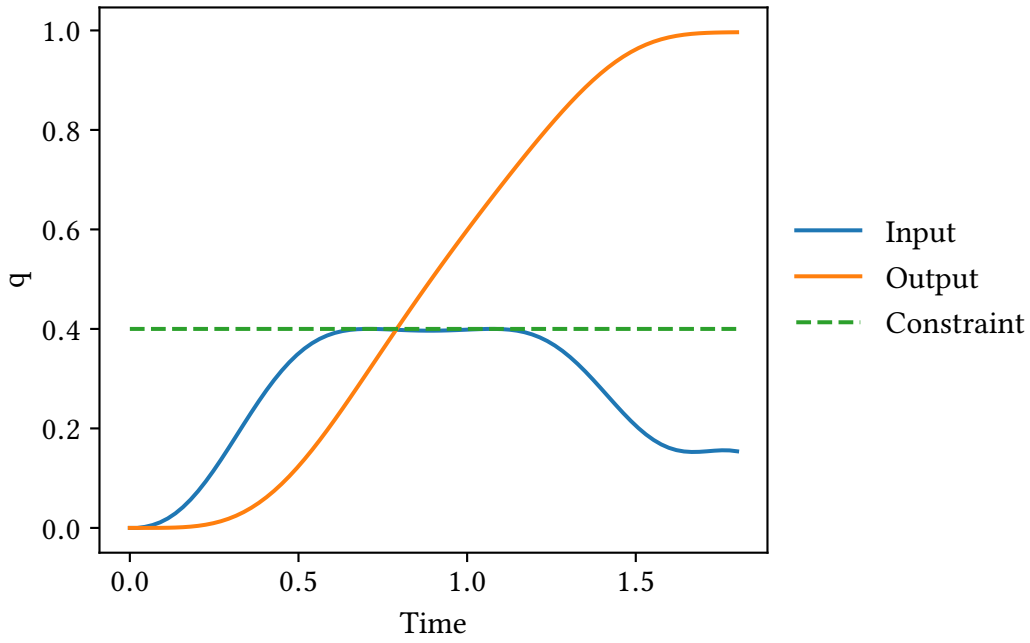


Figure 5.5: Input and output trajectory of scenario 3.

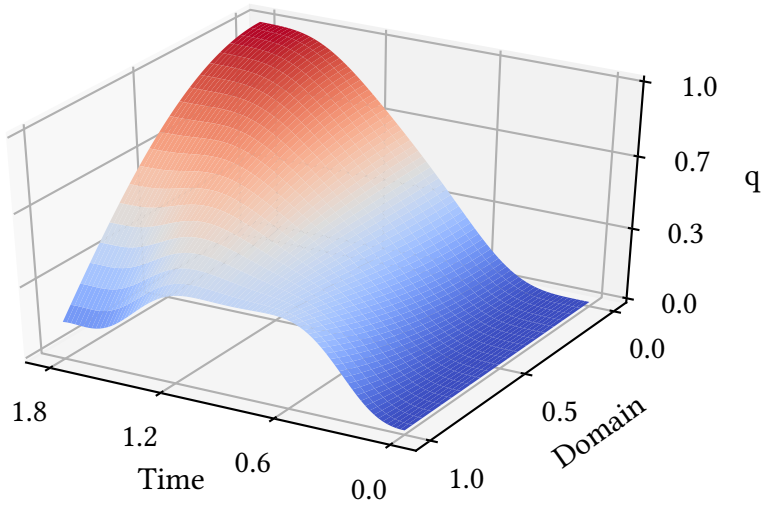


Figure 5.6: Profile of the solution of scenario 3.

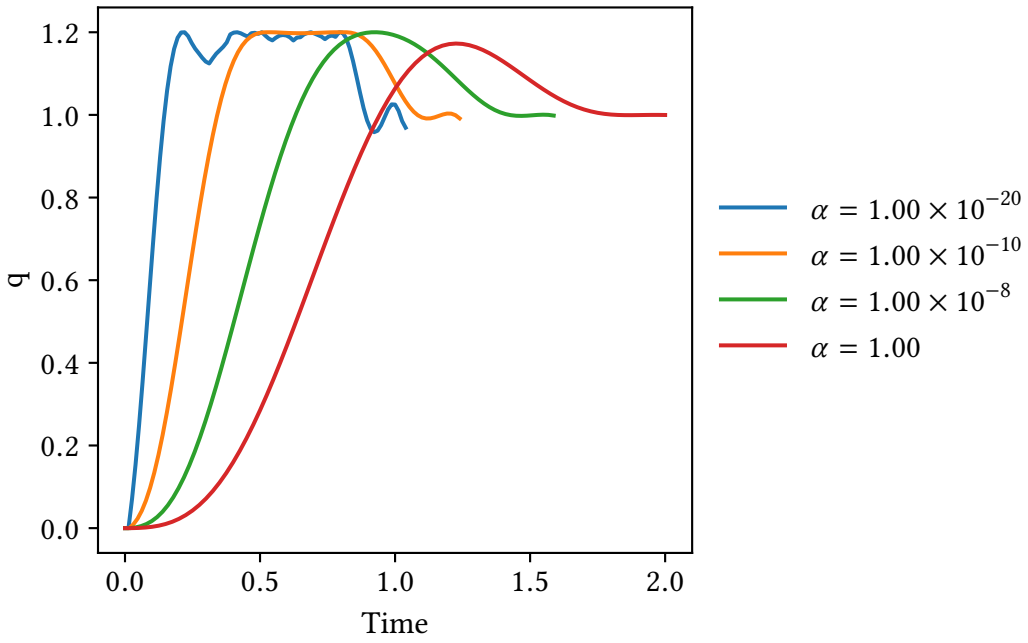


Figure 5.7: Input of scenario 1 with varying Tikhonov parameter.

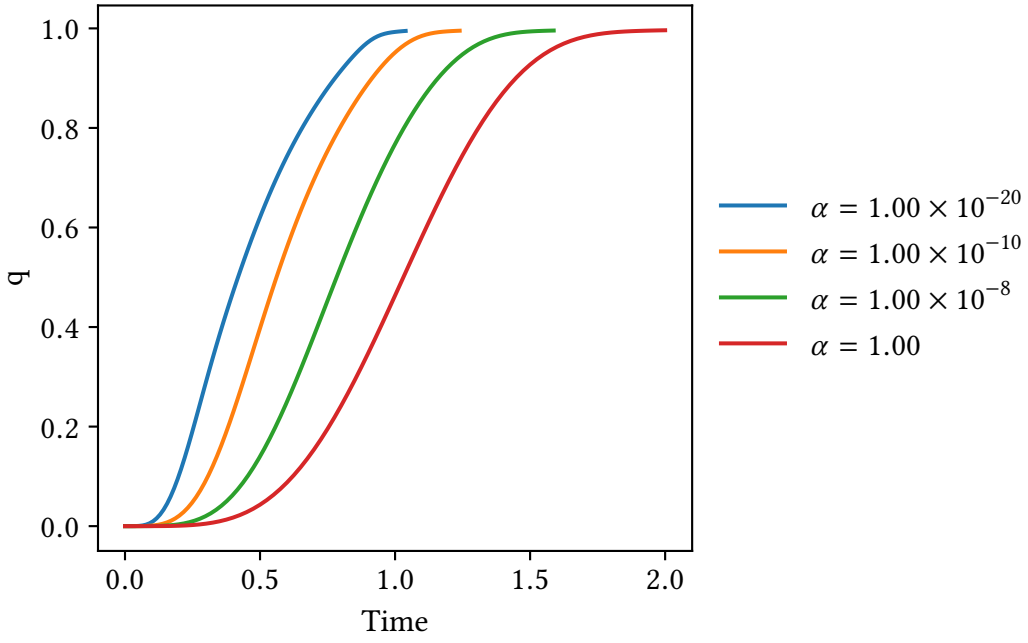


Figure 5.8: Output of scenario 1 with varying Tikhonov parameter.

5.2 Extension to flatness-based model predictive control

A suitable extension to the method derived in the previous section is the concept of model predictive control. Since Section 5.1 considers the constrained trajectory planning problem by computing a feed forward control input, the method is extended by employing a stabilizing model predictive control (MPC) formulation. A variety of academic and industrial applications for MPC can be found in [67, 35, 65]. General feedback controller designs are used in combinations with limiters to comply with restrictions on inputs or states. The MPC method can be appealing in those cases due to the fact that one can consider constraints on the control input and states in the formulation itself. Spatially discretized partial differential equations that form a system of ODEs can be used in combination with this approach but lead to high computational cost due to the optimization character of MPC. Therefore techniques that provide reduced order models like proper orthogonal decomposition (POD) to reduce the system size and computational burden are analyzed in [8, 2].

As in Section 5.1 the flatness-based method is used to parameterize the equation using a flat output to address the closed-loop by employing a MPC formulation. All derivations and results were published in [60].

5.2.1 Problem formulation

Equation (5.1) is extended to form a diffusion-convection-reaction system and a temporary state variable $\tilde{q}(z,t)$ is used to describe the problem, which is transformed later on in order to describe an equivalent formulation in $q(z,t)$ that has no convection term. Consider the domain $x \in (0,1)$ with

$$\frac{\partial \tilde{q}}{\partial t} = a\frac{\partial^2 \tilde{q}}{\partial x^2} + \nu\frac{\partial \tilde{q}}{\partial x} + r\tilde{q} \qquad \text{for } x \in (0,1)\,, t > 0, \tag{5.27a}$$

$$\frac{\partial \tilde{q}}{\partial x} = g(w(t) - \tilde{q}) \qquad \text{on } x = 0, \tag{5.27b}$$

$$\frac{\partial \tilde{q}}{\partial x} = b(\tilde{u} - \tilde{q}) \qquad \text{on } x = 1 \tag{5.27c}$$

and initial state $\tilde{x}(t_0) = \tilde{q}^0$. The parameters a, ν, r, g and b are for convenience to enable or disable certain effects or even disable contributions of, e.g., convection with $\nu = 0$. As mentioned above we transform the temporary state $\tilde{q}$ and input $\tilde{u}$ by substituting

$$\tilde{q} = \exp\left(-\frac{\nu}{2a}\right) q, \quad \tilde{u} = \exp\left(\frac{\nu}{2a}\right) u \tag{5.28}$$

in Equation (5.27a) and arrive at a diffusion-reaction system using $c = r - \frac{v^2}{4a}$

$$\frac{\partial q}{\partial t} = a\frac{\partial^2 q}{\partial x^2} + cq \qquad \text{for } x \in (0,1)\,, t > 0, \tag{5.29a}$$

$$\frac{\partial q}{\partial x} = gw(t) - \left(g - \frac{v}{2a}\right) q \qquad \text{on } x = 0, \tag{5.29b}$$

$$\frac{\partial q}{\partial x} = bu - \left(b - \frac{v}{2a}\right) q \qquad \text{on } x = 1 \tag{5.29c}$$

and initial state $q(t_0) = q^0$. A general MPC implementation looks like Algorithm 5 (see [35]). Here, $U \subset \mathbb{U}$ denotes the feasible input space and $Q \subset \mathbb{Q}$ the state space.

Algorithm 5 Model Predictive Control

Set horizon T
for $k = 1, 2, \ldots$ **do**
  $q^0 := q(t_k)$
  Solve the optimal control problem on the horizon T

$$\min_u J = \phi(q(t_k + T)) + \int_{t_k}^{t_k+T} l(q, u, t)\, dt$$

  subject to Equation (5.29) with $u \in U$ and $q \in Q$.
  Use $\bar{u}$ from the optimal control problem for the next control interval.
end for

This notation serves as a formulation for state and input constraints. Reviewing the MPC process, it can be observed that for every control interval an optimization problem defined over a prescribed horizon T has to be solved.

5.2.2 State and input parametrization

Similar to Section 5.1, a flat output to Equation (5.29) has to be derived. The procedure of formal integration was shown in [56, 75] and used in the following. Subsequently Equation (5.29) is solved for $\frac{\partial^2 q}{\partial x^2}$ and integrated twice with respect to the domain variable x which results in

$$q = y\left(1 + x\left(\frac{v}{2a} - g\right)\right) + gxw + \frac{1}{a}\int_0^x \int_0^p \left(\frac{\partial q}{\partial t} - cq\right) dxdp. \tag{5.30}$$

By using the infinite series approximation

$$q = \sum_{i=0}^{\infty} q_i \tag{5.31}$$

to substitute the state q in Equation (5.30) we get

$$q_0 = y\left(1 + x\left(\frac{\nu}{2a} - g\right)\right) + gxw, \tag{5.32}$$

$$q_{i+1} = \frac{1}{a}\int_0^x \int_0^p \left(\frac{\partial q}{\partial t} - cq\right) dsdp. \tag{5.33}$$

Every state coefficient q_i can be formally expressed by the flat output y and the already known function w thus the parameterized state coefficients are denoted by $q_i(y)$. The input u can be formulated in terms of y by

$$u = \frac{\exp\left(-\frac{\nu}{2a}\right)}{b} \sum_{i=0}^{\infty} \left(\frac{\partial q_i(y)}{\partial z} + \left(b - \frac{\nu}{2a} q_i(y)\right)\right) \quad \text{on } x = 1. \tag{5.34}$$

Like in Section 5.1 the integrator chain approach for the definition of y is used instead of incorporating a Gevrey class function so that the ODE system defined in Equation (5.12a) defines the flat output y and its derivatives. Therefore we define again

$$\begin{aligned} \dot{z} &= \underbrace{\begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{A} z + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{b} v \\ z(0) &= z^0 \in \mathbb{R}^{\beta} \end{aligned} \tag{5.35a}$$

with the state

$$z = \left[y, y^{(1)}, \dots, y^{(\beta-2)}, y^{(\beta-1)}\right]^T \tag{5.35b}$$

and input

$$v = y^{(\beta)}. \tag{5.35c}$$

As stated in Equation (5.12a) the variable v is the highest derivative of the β-times differentiable function y. Using Algorithm 5 with the parameterized system requires a reformulation of the objective functional J in terms of the flat output y described through z, which yields

$$\min J = \phi(z) + \int_{t_k}^{t_k+T} l(z, v, t)\, dt. \tag{5.36}$$

Additionally a state observer is designed by assuming that the flat output y can be measured. The fact that y is described by the integrator chain in Equation (5.12a), enables the possibility to reconstruct the whole state q through the observer described by

$$\dot{\hat{z}} = A\hat{z} + bv + L\tilde{z}_1 \tag{5.37a}$$

$$\tilde{z}_1 = c^T(z - \hat{z}) \tag{5.37b}$$

for $t > 0$ and $\hat{z}(0) = \hat{z}^0$ and $c^T = [1, 0, \ldots, 0]$. The observer gain is determined by the use of the Ackermann formula, which yields

$$L = \left(p_0\mathbb{I} + p_1 A + \cdots + p_{N-1}A^{N-1} + A^N\right) w,$$

with the coefficients of a Hurwitz polynomial p_i and w the last column of the inverse observability matrix.

5.2.3 Time discretization

For simulation purposes the equations involved to solve Equation (5.36) have to be discretized in time. We use Equation (5.31) for the evaluation of the spatial state q and stop the series at the upper limit β and define

$$\bar{q} = \sum_i^{\beta} q_i(v), \quad \bar{u} = \sum_i^{\beta} u_i(v). \tag{5.38}$$

Recalling Section 5.1.1 and the piecewise constant function for the highest derivative of the flat output

$$V_k = \begin{bmatrix} v_0 & v_1 & \ldots & v_{k-1} \end{bmatrix}^T \in \mathbb{R}^k,$$

we define the state $\bar{q}$ and input $\bar{u}$ at the time step t_k as

$$\bar{q}_k(V_k) = \bar{q}(v)|_{t=t_k}, \quad \bar{u}_k(V_k) = \bar{u}(v)|_{t=t_k}. \tag{5.39}$$

The optimal control problem on the time horizon T that has to be solved in every time step to determine the control for the next interval is determined by

$$\min_{V_k} J = \phi(\bar{q}_N(z_k, V_k)) + \int \sum_{k=0}^{N} a_k l(\bar{q}_k(z_k, V_k), \bar{u}_k(z_k, V_k))\, dx, \tag{5.40a}$$

subject to state and input constraints

$$q^- \leq \bar{q}_k(z_k, V_k) \leq q^+ \quad x \in [0, 1], \tag{5.40b}$$

$$u^- \leq \bar{u}_k(z_k, V_k) \leq u^+, \tag{5.40c}$$

for $k = 0, 1, \ldots, N$.

Remark 12. Note that every function in the objective functional and constraints is only dependent on the flat output and its derivatives. Therefore no evaluation of the partial differential equation is necessary for solving the optimization problem.

5.2.4 Examples

For the examples a direct approach is used assuming a full discretization of Equation (5.40). To implement Algorithm 5 we use the sampling interval $T_s = 0.01$ and evaluate Equation (5.29) in every time step to provide a measurement for the state observer. With the provided initial state $\hat{z}(t_k)$ at the beginning of the horizon, a solution to Equation (5.40) is computed over the interval $T = NT_s$. We use the objective functional

$$\begin{aligned}\min_u J = &\|\bar{q}_N(z_k, V_k)|_{x=0} - q^*\|_{L_2}^2 \\ &+ \sum_{k=0}^{N} a_k \left(\|\bar{q}_k(z_k, V_k)|_{x=0} - q^*\|_{L_2}^2 + \bar{u}(z_k, V_k)^2 \right),\end{aligned} \tag{5.41}$$

with the the first term weighing the terminal state $\bar{q}_N$ of the horizon. For the first scenario the parameters of Equation (5.29) are set to $a = 1, v = 2, c = 5, g = 1$ and $b = \infty$ such that the mixed boundary conditions yields an inhomogeneous Dirichlet boundary condition. The choice of the parameters yields an unstable partial differential equation when no stabilization controller is applied.

The first scenario assumes an initial condition of $q^0 = x - 0.5$, which leads to an unstable state, and the desired steady state is chosen to $q^* = 0$. The observer initial state is set to $\hat{z}^0 = 0$ which differs from q^0 and therefore the observer has to converge to the state z. All control inputs are illustrated in Figure 5.9. Results without constraints are shown in Figure 5.10. In Figure 5.11 only state constraints are considered with $-0.5 \leq z \leq 0.5$. In Figure 5.12 only input constraints are imposed by $-1 \leq u \leq 1$ and one can clearly observe that the input constrained control inputs are at the edge of feasibility for the chosen equation but still lead to a stabilized profile. In a second example scenario the initial condition $q^0 = 0$ is used and the desired solution is set to $q^* = 1$ at $x = 1$. The parameters are the same as in the first example. This scenario also demonstrates using an excitation signal $w = sin(2\pi t)$. In Figure 5.13 the computed control inputs are illustrated. In the example without excitation, the desired state is reached without much effort from the control input which can be observed in Figure 5.14. Reviewing Figure 5.15, the excitation signal leads to a persistent error in the desired profile and no steady state can be reached.

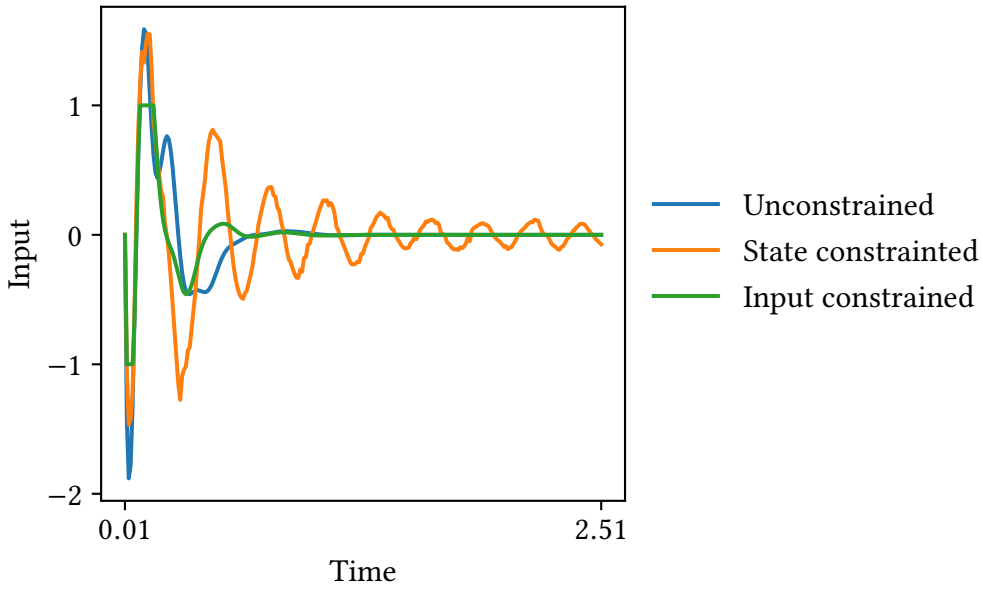


Figure 5.9: Controls for the MPC scenario with non-zero initial condition to zero steady state.

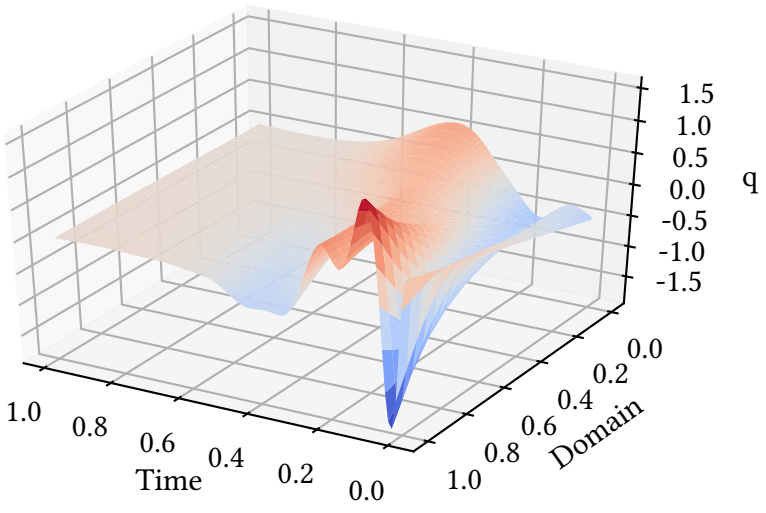


Figure 5.10: MPC scenario from non-zero initial condition to zero steady state without constraints.

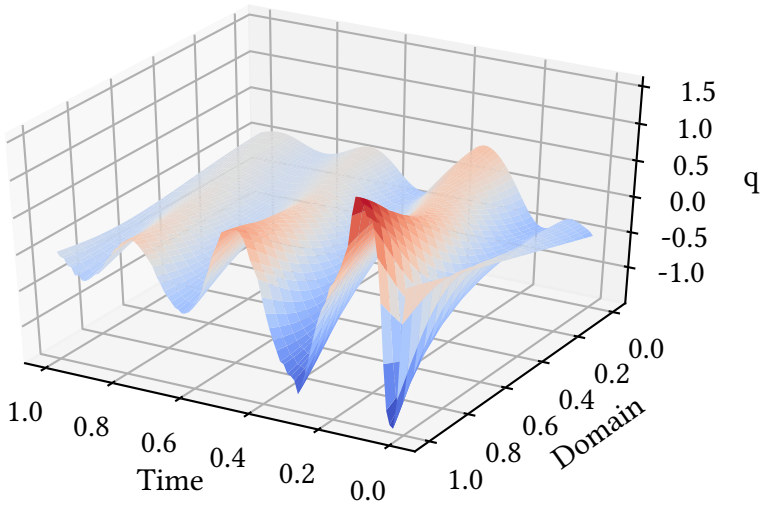


Figure 5.11: MPC scenario from non-zero initial condition to zero steady state with state constraints.

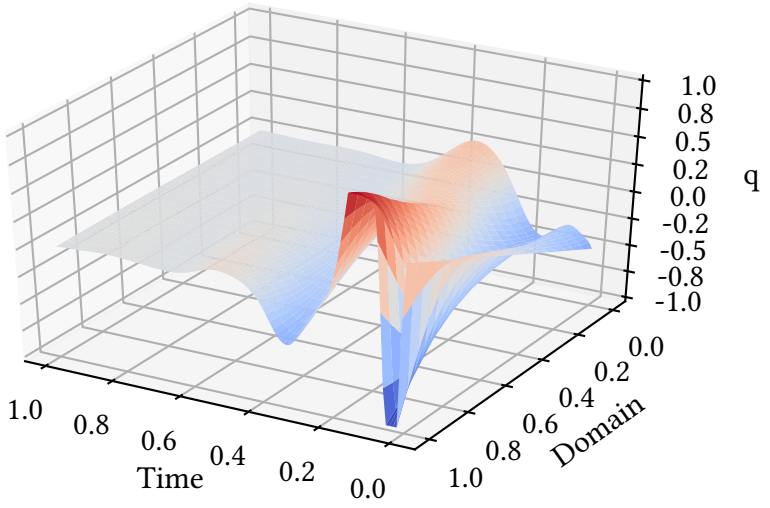


Figure 5.12: MPC scenario from non-zero initial condition to zero steady state with input constraints.

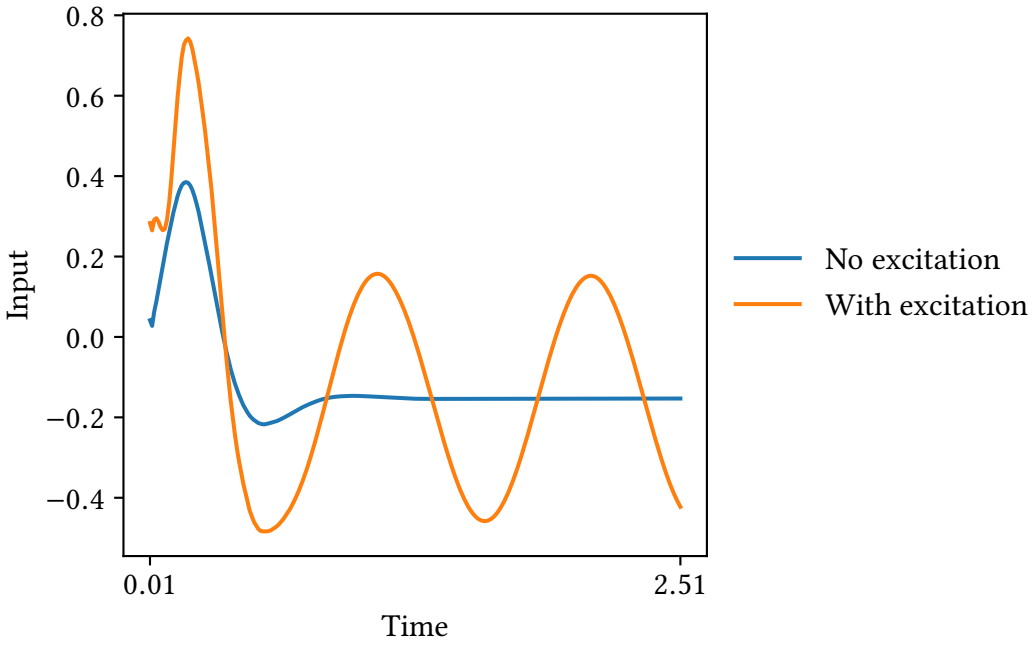


Figure 5.13: MPC scenario from zero initial condition to steady state.

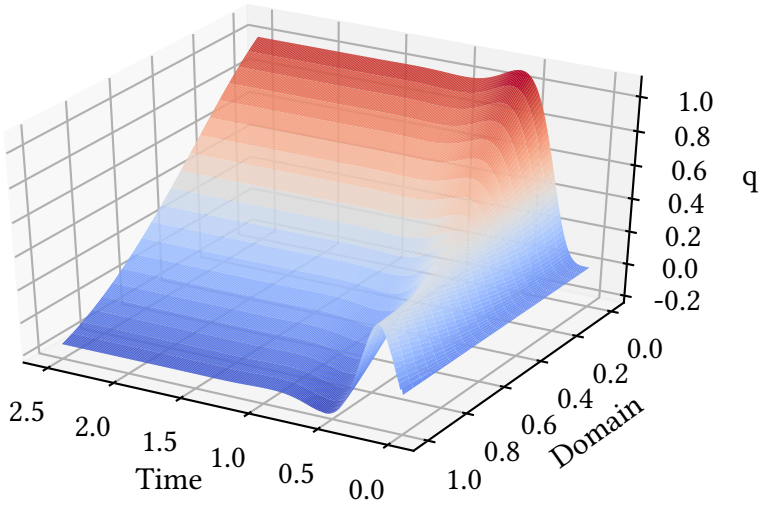


Figure 5.14: MPC scenario from zero initial condition to steady state without excitation.

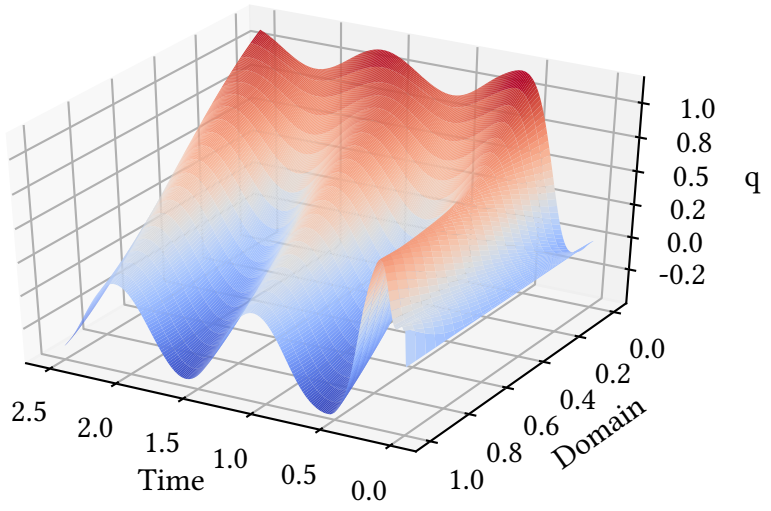


Figure 5.15: MPC scenario from zero initial condition to steady state with excitation.

Chapter 6

Conclusion

In this work a description to the solution of finite element method based discretizations for conductive and convective heat transfer is given to provide a major component for analysis of optimization problems leading to trajectory planning algorithms for optimal control. In Chapter 1 an introduction to the finite element method is given and followed by the derivation of conductive heat transfer through the heat equation in Chapter 2.

The physical phenomena are extended in Chapter 3 after summarizing the basics of fluid flow by deriving the Navier-Stokes equations. Both physics, the conductive and convective heat transfer, are discussed and an appropriate discretization is derived. Leveraging recent software libraries for discretization like `Firedrake` and `MFEM`, as well as `PETSc` and `hypre` for the solution of the arising linear and nonlinear systems, allows the efficient numerical solution of the mentioned equations. The implementation using these frameworks enabled a parallel computation which was crucial for the optimization problems which involved several solutions of the equations during the optimization process. There is also an emphasis on a detailed validation process to guarantee a flawless implementation which is carried out using the Method of Manufactured Solutions as well as several benchmark problems that provided public data for comparison.

In Section 3.4 the topic of multiphysics coupling is treated and different coupling techniques were presented. The arising equations from Chapter 3 are tightly coupled in the symbolic formulation. For optimization problems the equations, that have to be solved for objective and gradient evaluations, should be as simple as possible. Therefore a numerical comparison of the coupling methods is given to serve as a basis for argumentation on decoupling for optimization problems. These showed that it is feasible to ignore a tight coupling between fluid flow and energy transport in the numerical solution, if there is no necessity for higher order time stepping techniques.

In Chapter 4 an introduction on optimization concepts were given and fol-

lowed by a physical experiment that served as an example for a real world trajectory planning problem using the techniques of optimal control. A physical lab experiment was developed for this purpose, consisting of a heat control and monitoring system for an aluminum block. Tests were done using a desired profile consisting of a homogeneous and inhomogeneous temperature. Both examples showed efficient computational time for the trajectory planning problem considering the system size and both scenarios could be solved up to a certain error which correlated with the sensor error. The comparison of the simulations and measurements during the experiments were in good agreement as well.

Taking advantage of the observations in Section 3.4 a heating ventilation and air (HVAC) conditioning scenerio is shown in Section 4.3 to demonstrate the decoupling of physics and their feasibility for the optimization problems. A two dimensional problem that consists of heating up a room in a certain time frame using a variable temperature at an air conditioning inlet has been illustrated. Due to the formulation of the problem, only the energy equation part of the state equations is used for the optimization process. Two different scenarios are considered. In an idealized example a symmetric inlet is defined to heat up the room and the algorithm and optimization procedure resulted in a trajectory that controlled the room temperature to the desired state up to a small error in a narrow time frame. A more involving example for the control was shown using an asymmetric, smaller, inlet. Due to the resulting velocity field, this example gave a worse result considering the desired temperature distribution.

An alternative method using a novel approach, named flatness-based optimal control, was shown in Chapter 5. Here the flatness property of the PDE is used to derive an integrator chain representation of the flat output and detailed explanations about the algorithm along with numerical examples were presented. Developing this technique allow incorporation of input and state constraints in the trajectory planning using flatness-based methods. Furthermore it allows to neglect the actual PDE in the optimization process by using the integrator chain parameterization, which led to efficient computation times. Several examples involving pure diffusion and combined diffusion-reaction equations have been shown for numerical examples.

Finally in Section 5.2 this method is extended to a Model Predictive Control approach which allows stabilizing properties and a closed-loop analysis. Parameterizing the optimization problem with the integrator chain of the flat output provided a basis for the MPC formulation of a mixed diffusion-convection-reaction system. An example consisting of an unstable initial condition and a desired state of a zero profile is shown with an unconstrained, state constrained and input constrained formulation. A second example is provided to show the transition from a zero steady state to a desired steady state, once without any external disturbance and with an external disturbance on the state. All examples showed

desired behavior and promising results.

Finally, suggestions on further research regarding the topics is given. Considering the results provided by the optimal control problem formulation for the heating process of an aluminum block in Section 4.2, one should look at an improvement and extension of the lab experiment. The first addition could be a homogeneous test environment to guarantee a specific ambient temperature. Furthermore a cooling concept, consisting of either water cooling pipes or forced convection cooling through ventilation, is an interesting addition. This would enable to lower the bounds on the desired profile and control inputs for the optimization problem and allow for even more interesting profile selections. Beyond that a comprehensive analysis of feasibility to control and measure temperature gradients in the material is another interesting topic.

Evaluating the approaches of Section 4.3 lead to the conclusion that the method is interesting in terms of a priori analysis, but due to the long solution times, resulting from the needed discretizations, is not feasible in real time circumstances. An extension to three dimensions would increase the computational burden, but opens up room for deeper analysis in flow profiles and temperature distribution.

In Chapter 5 and Section 5.2 promising results regarding feasibility of the flatness-based optimal control were shown. Extending these methods to multi-dimensional approaches could show feasibility for more complex applications. Furthermore the application of the concept to provide a closed-loop control algorithm for the heating process of an aluminum block lab experiment would be a viable extension as well.

Appendices

Appendix A

Adjoint of conductive heat transfer

The transient heat equation with multiple actuators is described in Section 4.2.2 subsequently we present a derivation of the necessary adjoint and gradient formulation given

$$\rho C_p \frac{\partial T}{\partial t} - k\nabla^2 T = 0 \qquad \text{in } \Omega \times (t_0, t_e), \tag{A.1}$$

$$\boldsymbol{n} \cdot k\nabla T = c_i Q_i \qquad \text{on } \Gamma_{s_i} \times (t_0, t_e), \tag{A.2}$$

$$\boldsymbol{n} \cdot k\nabla T = \kappa(T_\infty - T) \qquad \text{on } \Gamma_c \times (t_0, t_e), \tag{A.3}$$

with the objective functional

$$J = \iint_\Omega \frac{\gamma_0}{2}(T - T^*)^2 \, dxdt + \sum_{i=1}^{N_a} \int \frac{\gamma_1}{2} c_i^2 \, dt + \int_\Omega \frac{\gamma_2}{2}(T(t_e) - T^*)^2 dx. \tag{A.4}$$

Equation (A.4) includes a tracking term and a final condition on the whole domain Ω instead of just a specific surface (e.g. Γ_{top}) in contrast to the objective functional presented in Section 4.2.2. The derivation of the adjoint equation and gradient does not differ because of this. We start with formulating the Lagrangian by adding the state equation Equation (A.1) to the objective function using the Lagrange multiplier λ, resulting in

$$\mathcal{L} = \underbrace{J}_{\mathcal{L}_1} + \underbrace{\iint_\Omega \rho C_p \frac{\partial T}{\partial t} \lambda \, dxdt - \iint_\Omega k\nabla^2 T \lambda \, dxdt}_{\mathcal{L}_2}. \tag{A.5}$$

Since c_i is the input variable which controls the amplitude of the actuator input we introduce the shorthand notation for a perturbation of c_i which results in a

variation of dependent variables like the temperature T and we define

$$\tilde{T} = T(c_i + h\hat{c}_i). \tag{A.6}$$

The two parts of the Lagrangian $\mathcal{L}_1$ and $\mathcal{L}_2$ are treated separately for clarity. Starting with the variation of $\mathcal{L}_1$ by perturbing c_i we get

$$\begin{aligned}\delta\mathcal{L}_{1_i} = \frac{\partial}{\partial h}\Bigg[\iint_\Omega \frac{\gamma_0}{2}(\tilde{T} - T^*)^2\,dxdt + \int \frac{\gamma_1}{2}(c_i + h\hat{c}_i)^2\,dt \\ + \iint_\Gamma \frac{\gamma_2}{2}(\tilde{T}(t_e) - T^*)^2\,dsdt\Bigg]_{h=0}.\end{aligned} \tag{A.7}$$

Note that the sum notation is not incorporated because we just look at one control variable for the derivation, since it is the same for each actuator. This fact results in the final gradient being a vector of size N_a in the continuous case. Defining the notation

$$\hat{T} = \frac{\partial}{\partial h}\Big[T(c_i + h\hat{c}_i)\Big]_{h=0}, \tag{A.8}$$

the variation for $\mathcal{L}_1$ in the i-th control c_i yields

$$\begin{aligned}\delta\mathcal{L}_{1_i} = \iint_\Omega \gamma_0(T - T^*)\hat{T}\,dxdt + \int \gamma_1 c_i\hat{c}_i\,dt \\ + \iint_\Omega \gamma_2(T(t_e) - T^*)\hat{T}(t_e)\,dxdt.\end{aligned} \tag{A.9}$$

Additionally the $\mathcal{L}_2$ has to treated in a similar way. Starting with the perturbation of the i-th control c_i and integrating the first term by parts in the time variable t results in

$$\begin{aligned}\delta\mathcal{L}_{2_i} = \frac{\partial}{\partial h}\Bigg[\Bigg[\int_\Omega \rho C_p\lambda\tilde{T}\,dx\Bigg]_{t_0}^{t_e} - \iint_\Omega \rho C_p\frac{\partial\lambda}{\partial t}\tilde{T}\,dxdt \\ + \iint_\Omega k\nabla\lambda\cdot\nabla\tilde{T}\,dxdt - \iint_\Gamma k\lambda\nabla\tilde{T}\cdot\boldsymbol{n}\,dsdt\Bigg]_{h=0}.\end{aligned} \tag{A.10}$$

Integrating by parts in the domain twice, using Greens identity we get

$$\begin{aligned}\delta\mathcal{L}_{2_i} = \frac{\partial}{\partial h}\Bigg[\Bigg[\int_\Omega \rho C_p\lambda\tilde{T}(t_e)\,dx - \int_\Omega \rho C_p\lambda\tilde{T}(t_0)\,dx\Bigg] - \iint_\Omega \rho C_p\frac{\partial\lambda}{\partial t}\tilde{T}\,dxdt \\ - \iint_\Omega k\nabla^2\lambda\tilde{T}\,dxdt + \iint_\Gamma k\tilde{T}\nabla\lambda\cdot\boldsymbol{n}\,dsdt - \iint_\Gamma k\lambda\nabla\tilde{T}\cdot\boldsymbol{n}\,dsdt\Bigg]_{h=0}.\end{aligned} \tag{A.11}$$

Now terms on the boundary can be replaced by the predefined boundary conditions

$$\begin{aligned}\delta\mathcal{L}_{2_i} = &\left[\int_\Omega \rho C_p \lambda \hat{T}(t_e)\,dx - \int_\Omega \rho C_p \lambda \hat{T}(t_0)\,dx\right] \\ &- \iint_\Omega \rho C_P \frac{\partial \lambda}{\partial t}\hat{T}\,dxdt - \iint_\Omega k\nabla^2\lambda\hat{T}\,dxdt \\ &+ \frac{\partial}{\partial h}\left[\iint_{\Gamma_{s_i}} k\nabla\lambda\cdot\boldsymbol{n}\tilde{T}\,dsdt - \iint_{\Gamma_{s_i}} \lambda Q_i(c_i + h\hat{c}_i)\,dsdt\right. \\ &\left.+ \iint_{\Gamma_c} k\nabla\lambda\cdot\boldsymbol{n}\tilde{T}\,dsdt - \iint_{\Gamma_c} \lambda\kappa(T_\infty - \tilde{T})\,dsdt\right]_{h=0}. \end{aligned} \tag{A.12}$$

Finally the derivative $\frac{\partial}{\partial h}$ is pulled into the remaining terms and the resulting equation yields

$$\begin{aligned}\delta\mathcal{L}_{2_i} = &\left[\int_\Omega \rho C_p \lambda \hat{T}(t_e)\,dx - \int_\Omega \rho C_p \lambda \hat{T}(t_0)\,dx\right] \\ &- \iint_\Omega \rho C_P \frac{\partial \lambda}{\partial t}\hat{T}\,dxdt - \iint_\Omega k\nabla^2\lambda\hat{T}\,dxdt \\ &+ \iint_{\Gamma_{s_i}} k\nabla\lambda\cdot\boldsymbol{n}\hat{T}\,dsdt - \iint_{\Gamma_{s_i}} \lambda Q\hat{c}_i\,dsdt \\ &+ \iint_{\Gamma_c} k\nabla\lambda\cdot\boldsymbol{n}\hat{T}\,dsdt + \iint_{\Gamma_c} \kappa\lambda\hat{T}\,dsdt. \end{aligned} \tag{A.13}$$

In order to fulfill the optimality conditions, the perturbed Lagrangian $\delta\mathcal{L}_i = \delta\mathcal{L}_{1_i} + \delta\mathcal{L}_{2_i}$ has to be zero. Therefore all terms containing $\hat{T}$ are collected. Looking at those terms and considering the integral over the domains and boundaries, this results in the adjoint state equation

$$-\rho C_p \frac{\partial\lambda}{\partial t} - k\nabla^2\lambda + \gamma_0(T - T^*) = 0 \qquad \text{in } \Omega\times(t_0,t_e), \tag{A.14}$$

$$k\nabla\lambda\cdot\boldsymbol{n} = 0 \qquad \text{on } \Gamma_{s_i}\times(t_0,t_e), \tag{A.15}$$

$$k\nabla\lambda\cdot\boldsymbol{n} + \kappa\lambda = 0 \qquad \text{on } \Gamma_c\times(t_0,t_e). \tag{A.16}$$

The resulting time dependent system in Equation (A.14) has a minus sign in front of the time dependent term. Therefore one has to integrate the system backwards in time or interchange the time integration bounds which results in a sign switch of the time dependent term. The initial condition (at t_e because of the backwards integration in time) can be derived from the remaining terms containing $\hat{T}(t_e)$

and

$$\rho C_p \lambda(t_e) + \gamma_2 (T(t_e) - T^*) = 0 \tag{A.17}$$

$$\lambda(t_e) = -\frac{\gamma_2}{\rho C_p} (T(t_e) - T^*) \tag{A.18}$$

Since there is no change of the initial condition in the state equation due to the perturbation of c_i, the relation

$$\int_\Omega \rho C_p \lambda \hat{T}(t_0)\, dx = 0$$

holds. Collecting all terms containing $\hat{c}_i$ results in the i-th gradient contribution of the objective functional

$$\frac{d\mathcal{L}_i}{dc_i} = \int \gamma_1 c_i \, dt - \iint_{\Gamma_{s_i}} \lambda Q_i \, dsdt. \tag{A.19}$$

Therefore the multi-dimensional gradient in continuous space

$$\frac{d\mathcal{L}}{dc} \in \mathbb{R}^{N_a} \quad \text{iff} \quad c \in \mathbb{R}^{N_a} \tag{A.20}$$

can be described by

$$\frac{d\mathcal{L}}{dc} = \begin{bmatrix} \int \gamma_1 c_1 \, dt - \iint_{\Gamma_{s_1}} \lambda Q \, dsdt \\ \int \gamma_1 c_2 \, dt - \iint_{\Gamma_{s_2}} \lambda Q \, dsdt \\ \vdots \\ \int \gamma_1 c_{N_a} \, dt - \iint_{\Gamma_{s_{N_a}}} \lambda Q \, dsdt \end{bmatrix}. \tag{A.21}$$

Remark 13. The whole derivation is in the continuous setting without any assumption of discretization in space or time. In Section 4.2.2 the problem is discretized in time with a first order BDF and therefore the resulting control inputs c_i are piecewise continuous over the time intervals. Assuming that there are N_t intervals the gradient for one control input c_i is of size $N_a \times N_t$.

For a more detailed discussion on variational methods in general see [69] as well as [53] for and optimization related discussion on Lagrange methods for general vector spaces.

Appendix B

Adjoint of convective heat transfer

In this section a comprehensive derivation of the adjoint equation and gradient formulation is given for the optimization problem described by Equation (4.28). Consider the energy equation part of the the problem described by

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \boldsymbol{u} \cdot \nabla T - k\nabla^2 T = 0 \qquad \text{in } \Omega \times (t_0, t_e), \tag{B.1}$$

$$\boldsymbol{n} \cdot k\nabla T = h(cQ - T) \qquad \text{on } \Gamma_s \times (t_0, t_e), \tag{B.2}$$

$$\boldsymbol{n} \cdot k\nabla T = 0 \qquad \text{on } \Gamma_w \cup \Gamma_o \times (t_0, t_e), \tag{B.3}$$

and the objective functional

$$J = \int \frac{\gamma_1}{2} c^2 \, dt + \int_\Omega \frac{\gamma_2}{2} (T(t_e) - T^*)^2 dx. \tag{B.4}$$

The Lagrangian is formed by adding the state equation Equation (B.1) to the objective functional and multiplying by the Lagrange multiplier λ and integration over the domain Ω and time interval t

$$\mathcal{L} = \underbrace{J}_{\mathcal{L}_1} + \underbrace{\iint_\Omega \rho C_p \frac{\partial T}{\partial t} \lambda \, dxdt + \iint_\Omega \rho C_p \boldsymbol{u} \cdot \nabla T \lambda \, dxdt - \iint_\Omega k\nabla^2 T \lambda \, dxdt}_{\mathcal{L}_2}. \tag{B.5}$$

As a shorthand notation we introduce

$$\tilde{T} = T(t, c + h\hat{c}), \tag{B.6}$$

to describe the effect of the perturbed independent variable c on a dependent variable. Forming the variation of $\mathcal{L}_1$ results in

$$\delta\mathcal{L}_1 = \frac{\partial}{\partial h}\left[\int \frac{\gamma_1}{2} (c + h\hat{c})^2 \, dt + \iint_\Gamma \frac{\gamma_2}{2} (\tilde{T}(t_e) - T^*)^2 \, dsdt \right]_{h=0} \tag{B.7}$$

and using the notation

$$\hat{T} = \frac{\partial}{\partial h}\Big[T(c + h\hat{c})\Big]_{h=0}, \tag{B.8}$$

the variation of $\mathcal{L}_1$ results in

$$\delta\mathcal{L}_1 = \int \gamma_1 c\hat{c}\, dt + \iint_\Omega \gamma_2 (T(t_e) - T^*)\hat{T}(t_e)\, dxdt. \tag{B.9}$$

The same procedure is used for the second term $\mathcal{L}_2$. Integration of terms by parts and applying the divergence theorem and Greens identity yields

$$\begin{aligned}\delta\mathcal{L}_2 = \frac{\partial}{\partial h}\Bigg[\Bigg[\int_\Omega \rho C_p \lambda\tilde{T}\, dx\Bigg]_{t_0}^{t_e} &- \iint_\Omega \rho C_p \frac{\partial\lambda}{\partial t}\tilde{T}\, dxdt \\ &- \iint_\Omega \rho C_p \boldsymbol{u}\cdot\nabla\lambda\tilde{T}\, dxdt + \iint_\Gamma \rho C_p \lambda\tilde{T}\boldsymbol{u}\cdot\boldsymbol{n}\, dsdt \\ &+ \iint_\Omega k\nabla\lambda\cdot\nabla\tilde{T}\, dxdt - \iint_\Gamma k\lambda\nabla\tilde{T}\cdot\boldsymbol{n}\, dsdt\Bigg]_{h=0}.\end{aligned} \tag{B.10}$$

To move all derivatives to the Lagrange multiplier λ the diffusion term has to be integrated by parts once more and we arrive at

$$\begin{aligned}\delta\mathcal{L}_2 = \frac{\partial}{\partial h}\Bigg[\Bigg[\int_\Omega \rho C_p \lambda\tilde{T}(t_e)\, dx - \int_\Omega \rho C_p \lambda\tilde{T}(t_0)\, dx\Bigg] &- \iint_\Omega \rho C_p \frac{\partial\lambda}{\partial t}\tilde{T}\, dxdt \\ &- \iint_\Omega \rho C_p \boldsymbol{u}\cdot\nabla\lambda\tilde{T}\, dxdt + \iint_\Gamma \rho C_p \lambda\tilde{T}\boldsymbol{u}\cdot\boldsymbol{n}\, dsdt \\ &- \iint_\Omega k\nabla^2\lambda\tilde{T}\, dxdt + \iint_\Gamma k\tilde{T}\nabla\lambda\cdot\boldsymbol{n}\, dsdt - \iint_\Gamma k\lambda\nabla\tilde{T}\cdot\boldsymbol{n}\, dsdt\Bigg]_{h=0}.\end{aligned} \tag{B.11}$$

Boundary terms can be replaced with the prescribed conditions from Equation (B.1) and therefore

$$\begin{aligned}\delta\mathcal{L}_2 = \Bigg[\int_\Omega \rho C_p \lambda\hat{T}(t_e)\, dx - \int_\Omega \rho C_p \lambda\hat{T}(t_0)\, dx\Bigg] &- \iint_\Omega \rho C_P \frac{\partial\lambda}{\partial t}\hat{T}\, dxdt \\ &- \iint_\Omega \rho C_p \boldsymbol{u}\cdot\nabla\lambda\hat{T}\, dxdt + \iint_\Gamma \rho C_p \lambda\hat{T}\boldsymbol{u}\cdot\boldsymbol{n}\, dsdt \\ &- \iint_\Omega k\nabla^2\lambda\hat{T}\, dxdt \\ &+ \frac{\partial}{\partial h}\Bigg[\iint_{\Gamma_s} k\nabla\lambda\cdot\boldsymbol{n}\tilde{T}\, dsdt - \iint_{\Gamma_s} \lambda h((c + h\hat{c})Q - T))\, dsdt\Bigg]_{h=0}.\end{aligned} \tag{B.12}$$

Finally the partial derivative is applied to the equation which yields

$$\begin{aligned}\delta\mathcal{L}_2 = &\left[\int_\Omega \rho C_p \lambda \hat{T}(t_e)\,dx - \int_\Omega \rho C_p \lambda \hat{T}(t_0)\,dx\right] - \iint_\Omega \rho C_P \frac{\partial \lambda}{\partial t}\hat{T}\,dxdt \\ &- \iint_\Omega \rho C_p \boldsymbol{u}\cdot\nabla\lambda\hat{T}\,dxdt + \iint_\Gamma \rho C_p \lambda \hat{T}\boldsymbol{u}\cdot\boldsymbol{n}\,dsdt \\ &- \iint_\Omega k\nabla^2\lambda\hat{T}\,dxdt + \iint_\Gamma \hat{T}k\nabla\lambda\cdot\boldsymbol{n}\,dsdt \\ &- \iint_{\Gamma_s} \lambda h Q\hat{c}\,dsdt + \iint_{\Gamma_s} \hat{T}\lambda h\,dsdt. \end{aligned} \tag{B.13}$$

The integral kernels have to eliminated and by collecting all terms containing $\hat{T}$ in the domain Ω and the boundary Γ the transient adjoint equation for the energy equation can be formulated

$$-\rho C_p \frac{\partial \lambda}{\partial t} - \rho C_p \boldsymbol{u}\cdot\nabla\lambda - k\nabla^2\lambda = 0 \qquad \text{in } \Omega\times(t_e,t_0), \tag{B.14}$$

$$(\rho C_p \lambda \boldsymbol{u} + k\nabla\lambda)\cdot\boldsymbol{n} + \lambda h = 0 \qquad \text{on } \Gamma_s\times(t_e,t_0), \tag{B.15}$$

$$(\rho C_p \lambda \boldsymbol{u} + k\nabla\lambda)\cdot\boldsymbol{n} = 0 \qquad \text{on } \Gamma_w\cup\Gamma_o\times(t_e,t_0). \tag{B.16}$$

The initial condition for the adjoint variable $\lambda(t_e)$ is obtained from terms containing $\hat{T}(t_e)$ and results in

$$\lambda(t_e) = -\frac{\gamma_2}{\rho C_p}(T(t_e) - T^*) \tag{B.17}$$

Collecting all terms containing the variation of the independent variable $\hat{c}$ leads to the gradient formulation

$$\frac{d\mathcal{L}}{dc} = \int \gamma_1 c\,dt - \iint_{\Gamma_s} \lambda h Q\,dsdt. \tag{B.18}$$

Bibliography

[1] V. Akçelik et al. “Parallel algorithms for PDE-constrained optimization”. In: *Parallel processing for scientific computing*. SIAM, 2006, pp. 291–322.

[2] A. Alla and S. Volkwein. “Asymptotic stability of POD based model predictive control for a semilinear parabolic PDE”. In: *Advances in Computational Mathematics* 41.5 (2015), pp. 1073–1102.

[3] J. D. Anderson and J. Wendt. *Computational fluid dynamics*. Vol. 206. Springer, 1995.

[4] J. Andrej and T. Meurer. “Flatness-based constrained optimal control of reaction-diffusion systems”. In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 2539–2544.

[5] S. Balay et al. “Efficient Management of Parallelism in Object Oriented Numerical Software Libraries”. In: *Modern Software Tools in Scientific Computing*. Ed. by E. Arge, A. M. Bruaset, and H. P. Langtangen. Birkhäuser Press, 1997, pp. 163–202.

[6] G. Bärwolff and M. Hinze. “Optimization of semiconductor melts”. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics* 86.6 (2006), pp. 423–437.

[7] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000.

[8] P. Benner, E. Sachs, and S. Volkwein. “Model order reduction for PDE constrained optimization”. In: *Trends in PDE constrained optimization*. Springer, 2014, pp. 303–326.

[9] P. B. Bochev, M. D. Gunzburger, and J. N. Shadid. “Stability of the SUPG finite element method for transient advection–diffusion problems”. In: *Computer Methods in Applied Mechanics and Engineering* 193.23-26 (June 2004), pp. 2301–2323.

[10] T. Böhm and T. Meurer. "Trajectory planning and tracking control for the temperature distribution in a deep drawing tool". In: *Control Engineering Practice* 64 (2017), pp. 127–139.

[11] J. Borggaard et al. "Control, estimation and optimization of energy efficient buildings". In: *American Control Conference, 2009. ACC'09.* IEEE. 2009, pp. 837–841.

[12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[13] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer New York, 2008.

[14] F. Brezzi. "On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers". en. In: *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 8.R2 (1974), pp. 129–151.

[15] A. N. Brooks and T. J. Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 32.1-3 (Aug. 1982), pp. 199–259.

[16] J. A. Burns, X. He, and W. Hu. "Control of the Boussinesq equations with implications for sensor location in energy efficient buildings". In: *American Control Conference (ACC), 2012*. IEEE. 2012, pp. 2232–2237.

[17] M. A. Christon, P. M. Gresho, and S. B. Sutton. "Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution)". In: *International Journal for Numerical Methods in Fluids* 40.8 (2002), pp. 953–980.

[18] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems (Classics in Applied Mathematics)*. SIAM: Society for Industrial and Applied Mathematics, 2002.

[19] S. D. Cohen, A. C. Hindmarsh, and P. F. Dubois. "CVODE, A Stiff/Nonstiff ODE Solver in C". In: *Computers in Physics* 10.2 (1996), p. 138.

[20] R. F. Curtain. "Pole assignment for distributed systems by finite-dimensional control". In: *Automatica* 21.1 (1985), pp. 57–67.

[21] H. Damanik et al. "A monolithic FEM-multigrid solver for non-isothermal incompressible flow on general meshes". In: *Journal of Computational Physics* 228.10 (June 2009), pp. 3869–3881.

[22] J. De Doná et al. “A flatness-based iterative method for reference trajectory generation in constrained NMPC”. In: *Nonlinear Model Predictive Control*. Springer, 2009, pp. 325–333.

[23] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press, 2002.

[24] W. B. Dunbar et al. “Motion planning for a nonlinear Stefan problem”. en. In: *ESAIM: Control, Optimisation and Calculus of Variations* 9 (2003), pp. 275–296.

[25] H. Elman et al. “Block Preconditioners Based on Approximate Commutators”. In: *SIAM Journal on Scientific Computing* 27.5 (Jan. 2006), pp. 1651–1668.

[26] K. Eppler and F. Tröltzsch. “Fast optimization methods in the selective cooling of steel”. In: *Online optimization of large scale systems*. Springer, 2001, pp. 185–204.

[27] D. Estep et al. “An a posteriori–a priori analysis of multiscale operator splitting”. In: *SIAM Journal on Numerical Analysis* 46.3 (2008), pp. 1116–1146.

[28] R. D. Falgout and U. M. Yang. “hypre: A Library of High Performance Preconditioners”. In: *Computational Science — ICCS 2002*. Ed. by P. M. A. Sloot et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 632–641.

[29] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2002.

[30] M. Fliess et al. “Flatness and defect of non-linear systems: introductory theory and examples”. In: *International journal of control* 61.6 (1995), pp. 1327–1361.

[31] D. R. Gaston et al. “Physics-based multiscale coupling for full core nuclear reactor simulation”. In: *Annals of Nuclear Energy* 84 (2015), pp. 45–54.

[32] P. E. Gill, W. Murray, and M. A. Saunders. “SNOPT: An SQP algorithm for large-scale constrained optimization”. In: *SIAM review* 47.1 (2005), pp. 99–131.

[33] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer Berlin Heidelberg, 1986.

[34] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms*. Vol. 5. Springer Science & Business Media, 2012.

[35] L. Grüne and J. Pannek. "Nonlinear model predictive control". In: *Nonlinear Model Predictive Control*. Springer, 2017, pp. 45–69.

[36] M. Gunzburger. *Perspectives in Flow Control and Optimization*. Society for Industrial and Applied Mathematics, 2002.

[37] M. D. Gunzburger and P. B. Bochev. *Least-Squares Finite Element Methods*. Springer New York, 2009.

[38] I. Hazyuk, C. Ghiaus, and D. Penhouet. "Optimal temperature control of intermittently heated buildings using Model Predictive Control: Part I–Building modeling". In: *Building and Environment* 51 (2012), pp. 379–387.

[39] I. Hazyuk, C. Ghiaus, and D. Penhouet. "Optimal temperature control of intermittently heated buildings using Model Predictive Control: Part II–Control algorithm". In: *Building and Environment* 51 (2012), pp. 388–394.

[40] R. He and H. Gonzalez. "Zoned HVAC control via PDE-constrained optimization". In: *American Control Conference (ACC), 2016*. IEEE. 2016, pp. 587–592.

[41] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[42] A. C. Hindmarsh et al. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers". In: *ACM Transactions on Mathematical Software* 31.3 (Sept. 2005), pp. 363–396.

[43] M. Hinze and U. Matthes. "Optimal and model predictive control of the boussinesq approximation". In: *Control of Coupled partial differential equations*. Springer, 2007, pp. 149–174.

[44] M. Hinze and T. Slawig. "Adjoint gradients compared to gradients from algorithmic differentiation in instantaneous control of the Navier-Stokes equations". In: *Optimization Methods and Software* 18.3 (2003), pp. 299–315.

[45] T. J. Hughes, L. P. Franca, and M. Balestra. "A new finite element formulation for computational fluid dynamics: V. Circumventing the babuška-brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations". In: *Computer Methods in Applied Mechanics and Engineering* 59.1 (1986), pp. 85–99.

[46] G. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, June 2005.

[47] C. T. Kelley. *Iterative methods for optimization*. Vol. 18. Siam, 1999.

[48] D. E. Keyes et al. “Multiphysics Simulations: Challenges and Opportunities”. In: *The International Journal of High Performance Computing Applications* 27.1 (2013), pp. 4–83.

[49] B. Kolar, H. Rams, and K. Schlacher. “Time-optimal flatness based control of a gantry crane”. In: *Control Engineering Practice* 60 (2017), pp. 18–27.

[50] M. Kronbichler and W. A. Wall. “A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers”. In: *SIAM Journal on Scientific Computing* 40.5 (2018), A3423–A3448.

[51] B. Laroche, P. Martin, P. Rouchon, et al. “Motion planning for the heat equation”. In: *International journal of robust and nonlinear control* 10.8 (2000), pp. 629–643.

[52] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Vol. 98. Siam, 2007.

[53] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.

[54] A. Lynch and J. Rudolph. “Flatness-based boundary control of a class of quasilinear parabolic distributed parameter systems”. In: *International Journal of Control* 75.15 (2002), pp. 1219–1230.

[55] P. Messina. “The Exascale Computing Project”. In: *Computing in Science Engineering* 19.3 (May 2017), pp. 63–67.

[56] T. Meurer and A. Kugi. “Trajectory Planning for Boundary Controlled Parabolic PDEs With Varying Parameters on Higher-Dimensional Spatial Domains”. In: *IEEE Transactions on Automatic Control* 54.8 (Aug. 2009), pp. 1854–1868.

[57] T. Meurer and M. Zeitz. “Feedforward and Feedback Tracking Control of Nonlinear Diffusion-Convection-Reaction Systems Using Summability Methods”. In: *Industrial & Engineering Chemistry Research* 44.8 (Apr. 2005), pp. 2532–2548.

[58] T. Meurer. *Control of Higher–Dimensional PDEs*. Springer Berlin Heidelberg, 2013.

[59] T. Meurer. “Flatness-based trajectory planning for diffusion–reaction systems in a parallelepipedon—A spectral approach”. In: *Automatica* 47.5 (2011), pp. 935–949.

[60] T. Meurer and J. Andrej. “Flatness–based model predictive control of linear diffusion–convection–reaction processes”. In: *2018 Conference on Decision and Control (CDC)*. Accepted for Publication. IEEE. 2018.

[61] T. Meurer and M. Zeitz. "Two-degree-of-freedom tracking control design for boundary controlled distributed parameter systems using formal power series". In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on.* Vol. 3. IEEE. 2004, pp. 3307–3312.

[62] *MFEM: Modular Finite Element Methods Library.* `mfem.org`.

[63] J. Oliver and C. A. D. Saracibar. *Continuum Mechanics for Engineers. Theory and Problems (First edition, September 2016).* Open Access, 2016.

[64] N. Petit and P. Rouchon. "Dynamics and solutions to some control problems for water-tank systems". In: *IEEE Transactions on Automatic Control* 47.4 (2002), pp. 594–609.

[65] S. J. Qin and T. A. Badgwell. "A survey of industrial model predictive control technology". In: *Control engineering practice* 11.7 (2003), pp. 733–764.

[66] F. Rathgeber et al. "Firedrake: automating the finite element method by composing abstractions". In: *ACM Transactions on Mathematical Software (TOMS)* 43.3 (2017), p. 24.

[67] J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design.* Nob Hill Pub. Madison, Wisconsin, 2009.

[68] R. Rebarber. "Spectral assignability for distributed parameter systems with unbounded scalar control". In: *SIAM journal on control and optimization* 27.1 (1989), pp. 148–169.

[69] J. N. Reddy. *Energy principles and variational methods in applied mechanics.* John Wiley & Sons, 2017.

[70] I. M. Ross and F. Fahroo. "Pseudospectral methods for optimal motion planning of differentially flat systems". In: *IEEE Transactions on Automatic Control* 49.8 (2004), pp. 1410–1413.

[71] Y. Saad. *Iterative methods for sparse linear systems.* Vol. 82. siam, 2003.

[72] K. Salari and P. Knupp. *Code Verification by the Method of Manufactured Solutions.* Tech. rep. Sandia National Laboratories, 2000.

[73] M. Schäfer et al. "Benchmark Computations of Laminar Flow Around a Cylinder". In: *Flow Simulation with High-Performance Computers II* (1996), pp. 547–566.

[74] R. Schmidt et al. "An approach for coupled-code multiphysics core simulations from a common input". In: *Annals of Nuclear Energy* 84 (2015), pp. 140–152.

[75] B. Schörkhuber, T. Meurer, and A. Jüngel. "Flatness of Semilinear Parabolic PDEs—A Generalized Cauchy–Kowalevski Approach". In: *IEEE Transactions on Automatic Control* 58.9 (Sept. 2013), pp. 2277–2291.

[76] J. Schröck, T. Meurer, and A. Kugi. "Motion Planning for Piezo-Actuated Flexible Structures: Modeling, Design, and Experiment". In: *IEEE Transactions on Control Systems Technology* 21.3 (May 2013), pp. 807–819.

[77] T. Schwedes et al. "Mesh dependence in PDE-constrained optimisation". In: *Mesh Dependence in PDE-Constrained Optimisation*. Springer, 2017, pp. 53–78.

[78] R. Serban and A. C. Hindmarsh. "CVODES: The Sensitivity-Enabled ODE Solver in SUNDIALS". In: *Volume 6: 5th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B, and C*. ASME, 2005.

[79] M. Staudecker, K. Schlacher, and R. Hansl. "Passivity Based Control and Time Optimal Trajectory Planning of a Single Mast Stacker Crane". In: *IFAC Proceedings Volumes* 41.2 (2008), pp. 875–880.

[80] C. Taylor and P. Hood. "A numerical solution of the Navier-Stokes equations using the finite element technique". In: *Computers & Fluids* 1.1 (1973), pp. 73–100.

[81] T. Tezduyar. "Stabilized Finite Element Formulations for Incompressible Flow Computations". In: *Advances in Applied Mechanics*. Elsevier, 1991, pp. 1–44.

[82] J. W. Thomas. *Numerical partial differential equations: finite difference methods*. Vol. 22. Springer Science & Business Media, 2013.

[83] F. Tröltzsch. *Optimal Control of Partial Differential Equations*. American Mathematical Society, Apr. 2010.

[84] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (Apr. 2005), pp. 25–57.

[85] J. H. Whitelaw. "Convective Heat Transfer". In: *A-to-Z Guide to Thermodynamics, Heat and Mass Transfer, and Fluids Engineering* (2011).

[86] J. Wloka. *Partielle Differentialgleichungen: Sobolevräume und Randwertaufgaben*. Teubner, 1982.

[87] F. Woittennek and J. Rudolph. "Motion planning for a class of boundary controlled linear hyperbolic PDE's involving finite distributed delays". In: *ESAIM: Control, Optimisation and Calculus of Variations* 9 (2003), pp. 419–435.

[88] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier LTD, Oxford, 2013.