A Branch-and-Bound Approach to Mixed-Integer Optimal Control Using POD

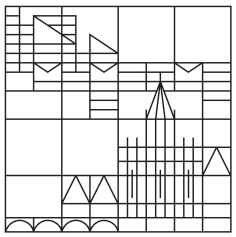
Master thesis

submitted by

Freya Bachmann

at the

Universität Konstanz



Department of Mathematics and Statistics

first reviewer: Prof. Dr. Stefan Volkwein
second reviewer: Jun.-Prof. Dr. Gabriele Ciaramella

Konstanz, July 2017

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der numerischen Lösung eines gemischt-ganzzahligen Optimalsteuerungsproblems (mixed-integer optimal control problem). Dieses wird durch ein physikalisches Beispiel motiviert: die äußeren Wände eines Raumes sollen mit Isoliermaterial ausgestattet werden und eine Fußbodenheizung so gesteuert werden, dass eine gewünschte Temperatur möglichst effizient angenähert wird. Die Wahl der Materialien erfolgt aus einer Menge an diskreten, ganzzahligen Randkontrollen, wohingegen die Heizung eine verteilte und kontinuierliche Kontrolle darstellt. Die Temperaturverteilung im Raum wird durch die lineare Wärmeleitungsgleichung mit Neumann-Randbedingungen beschrieben.

Für die gemischt-ganzzahlige Optimierung wenden wir den sogenannten branch-and-bound Algorithmus an. Um eine ganzzahlige optimale Randsteuerung zu finden, löst dieser iterativ relaxierte Probleme (d.h. ohne Ganzzahligkeitsbedingung) für verschiedene obere und untere Schranken an die Randsteuerung. Hierzu muss stets ein restringiertes linear-quadratisches Optimalsteuerungsproblem gelöst werden, weshalb insgesamt viele Zustands- und adjungierte Gleichungen gelöst werden müssen. Die Diskretisierung durch finite Elemente führt oft zu einer sehr großen Anzahl von Freiheitsgraden, die die Berechnung von Zustand und Adjungierter zeitintensiv machen. Daher wenden wir eine Modellreduktion mithilfe der POD-Methode an, die bei sehr kleinen Fehlern eine deutliche Beschleunigung auch in den verhältnismäßig unkomplizierten Beispielen zeigt.

Wir diskutieren verschiedene Strategien im branch-and-bound Algorithmus und zeigen mehrere numerische Beispiele für unterschiedliche Außentemperaturen.

Contents

1 Introduction

In times of increased environmental awareness and shrinking resources of fossil fuels one might imagine the importance of energy-efficient building operations. Thereby, our aim is to choose suitable insulation materials for the outer walls and determine an underfloor heating strategy to maintain a desired temperature in a room while keeping the costs reasonably low.

Mathematically, this translates into a mixed-integer optimal control problem which combines two major fields, mixed-integer programming (for which we will present the branch-and-bound algorithm) and optimal control (which we will speed-up by a reduced-order model approach).

Mixed-integer nonlinear programming (MINLP) has many applications in engineering, operations research and science – for instance in industrial production planning, in scheduling public transportation networks or in designing telecommunication networks. It involves discrete decisions that affect nonlinear system dynamics and so the final outcome. Thus, we have to face the combinatorial challenge of optimizing over a discrete variable set while managing nonlinear problems.
Most methods for solving MINLPs follow a tree-search of which we will present a classical single-tree method: the branch-and-bound algorithm. Other methods can be found in [1].

The system dynamics in the case we study (temperature distribution depending on the control) is governed by a linear-quadratic optimal control problem, so we are actually avoiding the difficulties arising from nonlinearities.

Optimal control is a frequent task in industry, engineering and science. The behaviour of a system which is in many cases governed by partial differential equations can be influenced by a control. The goal is to find an optimal control that induces a certain desired state of the system, often weighting against spending as little effort as possible.
Usually their discretization, e.g. by finite elements, gives large scale problems making the numerical optimization time-consuming. Hence, reduced-order models are a very attractive approach as they try to capture main

characteristics of the system dynamics. Using proper orthogonal decomposition (POD) we will compute specific ansatz functions leading to significantly smaller degrees of freedom in the finite element (FE) method.

We will shortly sketch the outline of the thesis. Chapter 2 is organized in five sections, first we will state the mixed-integer optimal control problem. In Section 2.2 we will discuss the unique solvability of the state equation and reformulate the minimization problem such that it purely depends on the controls. In the next section we will examine the existence of an optimal control and then derive first-order necessary optimality conditions. We will then present the FE method for tackling the problem numerically. The last section is dedicated to deriving a general a-posteriori estimate.

Chapter 3 addresses the model reduction. We present the continuous and discrete POD variant in the first two sections before deriving a low-dimensional model for the state and adjoint equation with the POD Galerkin ansatz.
The subsequent Chapter 4 is devoted to the branch-and-bound algorithm for solving mixed-integer problems. It explains the algorithm in general and gives strategies for branching decisions and node selection.

For different scenarios we will present numerical experiments in Chapter 5 and interpret the results. Finally, we draw conclusions and give an outlook to further interesting work in the last Chapter 6.

2 Problem formulation

In this chapter we will state the mixed-integer optimal control problem we are aiming to solve. We want to find an optimal solution for the two contrary tasks of achieving a desired temperature within a (horizontal cross section of a) room as closely as possible on the one hand while spending as little as possible on heating and insulation material on the other. The following sections will provide the existence of a global optimal control and derive first-order necessary optimality conditions. To treat the problem numerically we will discuss the finite element discretization and a posteriori error estimates. A good introduction to optimal control problems is given in [10].

For $T > 0$ let $[0, T] \subset \mathbb{R}$ be a time horizon and let $\Omega \subset \mathbb{R}^2$ be an open and bounded domain with Lipschitz-continuous boundary $\Gamma := \partial\Omega$. We allow a disjoint subsplitting of both the domain $\overline{\Omega} = \overline{\bigcup_{i=1}^{n} \Omega_i}$ into $n \in \mathbb{N}$ subdomains indicating individual underfloor heating tiles and of the boundary $\Gamma = \bigcup_{i=0}^{m} \Gamma_i$. Here, Γ_0 corresponds to the set of all interior walls whereas each boundary segment $\Gamma_1, \dots, \Gamma_m$ is an exterior wall or a window which shall be equipped with insulation material.
The subsplitting will be realized by shape functions $\chi_i^C \in L^\infty(\Omega)$ and $\chi_i^I \in L^\infty(\Gamma)$ which take non-zero values only on the corresponding subdomain Ω_i $(i = 1, \dots, n)$ or boundary segment Γ_i $(i = 1, \dots, m)$, respectively. We will consider the following optimal control problem:

$$\begin{aligned} \min\ J(y, u^C, u^I) := \frac{\alpha_Q}{2} \int_0^T \int_\Omega (y(t,x) - y_d(t,x))^2 \,\mathrm{d}x\,\mathrm{d}t \\ + \frac{1}{2} \sum_{i=1}^{n} \alpha_i^C \int_0^T u_i^C(t)^2 \,\mathrm{d}t + \frac{1}{2} \sum_{i=1}^{m} \alpha_i^I \,(u_i^I - \omega_i)^2 \end{aligned} \tag{2.1}$$

The system has two types of control variables, the distributed and time-dependent $u^C \in L^2(0, T; \mathbb{R}^n)$ representing the heating tiles and the time-independent boundary controls $u^I \in \mathbb{Z}^m$ denoting a certain choice of insulation material for $\Gamma_1, \dots, \Gamma_m$.

Remark 1. For simplicity these u^I shall be integer-valued. Actually, each material will come with a thermal transmittance coefficient describing the heat

transition in the heat equation which we will state in a moment and a certain price in the cost functional (also depending on the length of the boundary segment). But, as there will be a discrete and finite set of possible choices one could easily include such a mapping. ◊

This motivates the definition of the control space $U := L^2(0,T;\mathbb{R}^n) \times \mathbb{R}^m$ as a product of Hilbert spaces endowed with the standard product topology giving the norm

$$\begin{aligned}\|u\|_U &= \left(\left\|u^C(t)\right\|^2_{L^2(0,T;\mathbb{R}^n)} + \left\|u^I\right\|^2_{\mathbb{R}^m}\right)^{\frac{1}{2}} \\ &= \left(\int_0^T \left\|u^C(t)\right\|^2_{\mathbb{R}^n} \,\mathrm{d}t + \left\|u^I\right\|^2_{\mathbb{R}^m}\right)^{\frac{1}{2}}\end{aligned}$$

For $u, v \in U$ we define the componentwise comparison:

$$u \leq v \iff \begin{cases} u_i^C(t) \leq v_i^C(t) & 1 \leq i \leq n \text{ a.e. in } [0,T] \\ u_j^I \leq v_j^I & 1 \leq j \leq m \end{cases}$$

For both the distributed and the boundary control we introduce additional bilateral constraints $u_a, u_b \in U$ with $u_a \leq u_b$ such that

$$U_{ad}^{int} = U_{ad} \cap U^{int} \tag{2.2}$$

is non-empty where

$$U_{ad} = \{u \in U \mid u_a \leq u \leq u_b\}$$

denotes the set of admissible controls and the integrality constraint is ensured by

$$U^{int} = \{u = (u^C, u^I) \in U \mid u^I \in \mathbb{Z}^m\}$$

It is then our goal to find an optimal control

$$\bar{u} \in U_{ad}^{int} \tag{2.3}$$

in order to approach a desired inside temperature $y_d \in L^2(0,T;L^2(\Omega))$ while keeping the necessary heating and insulation costs reasonably low. The costs can be weighted against each other by coefficients $\alpha_Q, \alpha_1^C, \ldots, \alpha_n^C, \alpha_1^I, \ldots, \alpha_m^I \in \mathbb{R}_+ := \{x \in \mathbb{R} \,|\, x > 0\}$ and $\omega \in \mathbb{Z}^m$ is a reference control whose benefits will become apparent after introducing the heat equation now.

The state variable y in (2.1) describes the temperature at a time t and space x inside the room and is modelled by the linear heat equation with a source term depending on u^C and the thermal diffusivity coefficient $c \in \mathbb{R}_+$. For the interior walls we assume similar temperatures on both sides of the walls, thus neglectible heat transition which translates to homogeneous Neumann boundary conditions on Γ_0. Note that for all shape functions holds $\chi_i^I = 0$ $(i = 1, \dots, m)$ on Γ_0.

$$\begin{aligned} y_t(t,x) - c\,\Delta y(t,x) &= \sum_{i=1}^{n} u_i^C(t)\,\chi_i^C(x) && \text{for } (t,x) \in (0,T) \times \Omega \\ c\,\frac{\partial y}{\partial n}(t,s) &= \sum_{i=1}^{m} u_i^I\,(y_b - y_a(t))\,\chi_i^I(s) && \text{for } (t,s) \in (0,T) \times \Gamma \\ y(0,x) &= y_0(x) && \text{for } x \in \Omega \end{aligned} \tag{2.4}$$

For the exterior walls or windows as on $\Gamma_1, \dots, \Gamma_m$ we impose inhomogenous Neumann boundary conditions modelling the heat transition. It depends on the temperature difference which is modelled by a fixed value $y_b \in \mathbb{R}$ and a time-dependent outside temperature function y_a.

Remark 2. By this we avoid nonlinear constraints which would arise by taking the actual inside temperature y instead of y_b. Hence, y_b should give a suitable approximation to y everywhere and at all times. ◊

Furthermore, the heat transition in (2.4) depends on the heat transition coefficient given by a certain choice of insulation material u^I. Small values of u_i^I for $1 \le i \le m$ mean little heat transition and thus a good and usually more expensive insulation. The larger such a u_i^I for $1 \le i \le m$ the more heat will be lost and thus the worse and probably cheaper the insulation is. We model this inverse proportionality by subtracting the reference control ω in (2.1) which will be the cheapest available choice of insulation material: the upper bound u_b^I.

Remark 3. The cost functional J is strictly convex which follows directly from applying Young's inequality to all three summands. ◊

Finally, $y_0 \in L^2(\Omega)$ shall be some initial inside temperature.

We call (2.1)-(2.4) a mixed-integer optimal control problem (MIOCP).

2.1 The state variable

In order to state the weak formulation of the partial differential equation (PDE) (2.4) we will use results from functional analysis and the theory of partial differential equations which can be found for example in [3] and [4]. We introduce the Hilbert spaces $H = L^2(\Omega)$ and $V = H^1(\Omega)$ endowed with their standard inner products

$$\langle \varphi, \phi \rangle_H = \int_\Omega \varphi\phi \, \mathrm{d}x, \quad \langle \varphi, \phi \rangle_V = \int_\Omega \varphi\phi + \nabla\varphi \cdot \nabla\phi \, \mathrm{d}x$$

and their induced norms, respectively. Identifying H with its dual H' by the Riesz isomorphism yields a Gelfand triple $V \hookrightarrow H = H' \hookrightarrow V'$ with each embedding being continuous and dense. By V' we mean the topologic dual space of V, i.e. all linear and continuous functionals mapping from V to $\mathbb{R}$, thus in our setting $V' = H^{-1}(\Omega)$. The dual pairing is then denoted by $\langle \cdot\,, \cdot \rangle_{V',V}$.

Definition 2.1. *For $T > 0$ we define the space*

$$W(0,T) = \{y \in L^2(0,T;V) \,|\, y_t \in L^2(0,T;V')\}$$

where y_t denotes the weak derivative of y. It is a Hilbert space endowed with the inner product

$$\langle y, \phi \rangle_{W(0,T)} = \int_0^T \langle y, \phi \rangle_V + \langle y_t, \phi_t \rangle_{V'} \, \mathrm{d}t$$

The inner product in V' is given as the inner product of the Riesz representatives in V.

We list some helpful properties of $W(0,T)$, cf. [2] or Section 3.4 in [10].

- There exists a continuous embedding $W(0,T) \hookrightarrow C([0,T];H)$, i.e. a function $y \in W(0,T)$ is – after eventual modification on a set of measure zero – continuous w.r.t. time, so $y(0)$ and $y(T)$ are indeed meaningful.
- For all $y, \phi \in W(0,T)$ holds the integration by parts formula

  $$\int_0^T \langle y_t(t), \phi(t) \rangle_{V',V} + \langle \phi_t(t), y(t) \rangle_{V',V} \, \mathrm{d}t = \langle y(T), \phi(T) \rangle_H - \langle y(0), \phi(0) \rangle_H$$

- For all $y \in W(0,T)$ and $\varphi \in V$ holds

  $$\langle y_t(t), \varphi \rangle_{V',V} = \frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), \varphi \rangle_H$$

We will see that $W(0,T)$ is the appropriate space for the state variable.

For deriving the weak formulation of (2.4) we apply the standard procedure, i.e. assuming a classical solution, multiplying the equation by a testfunction $\varphi \in V$ and integrating over the domain. We cast a glance at the Laplacian term where we used Green's first identity and plugged in the boundary conditions:

$$\begin{aligned} -c\int_\Omega \Delta y\,\varphi \,\mathrm{d}x &= c\int_\Omega \nabla y\cdot\nabla\varphi \,\mathrm{d}x - c\int_{\partial\Omega}\varphi\frac{\partial y}{\partial n}\,\mathrm{d}S \\ &= c\int_\Omega \nabla y\cdot\nabla\varphi \,\mathrm{d}x - c\int_\Gamma \sum_{i=1}^m u_i^I\,(y_b - y_a(t))\,\chi_i^I(s)\,\varphi(s)\,\mathrm{d}s \end{aligned}$$

The weak formulation hence looks as follows:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega y(t)\varphi\,\mathrm{d}x + c\int_\Omega \nabla y(t)\cdot\nabla\varphi\,\mathrm{d}x &= \sum_{i=1}^n u_i^C(t)\int_\Omega \chi_i^C\varphi\,\mathrm{d}x \\ &\quad + \sum_{i=1}^m u_i^I(y_b - y_a(t))\int_\Gamma \chi_i^I\varphi\,\mathrm{d}S \\ \langle y(0),\varphi\rangle_H &= \langle y_0,\varphi\rangle_H \ \ \forall\varphi\in H \end{aligned}$$

We introduce the symmetric bilinear form

$$a\colon V\times V\to\mathbb{R},\quad a(\varphi,\phi) = c\int_\Omega \nabla\varphi\cdot\nabla\phi\,\mathrm{d}x \tag{2.5}$$

Proposition 2.2. *To the bilinear form a in (2.5) exist constants $\gamma,\gamma_1,\gamma_2>0$ such that for all $\varphi,\phi\in V$ holds:*

$$\begin{aligned} |a(\varphi,\phi)| &\le \gamma\,\|\varphi\|_V\,\|\phi\|_V \\ a(\varphi,\varphi) &\ge \gamma_1\,\|\varphi\|_V^2 - \gamma_2\,\|\varphi\|_H^2 \end{aligned} \tag{2.6}$$

that means a is bounded and coercive.

Proof. By the Cauchy-Schwarz inequality and $\varphi^2,\phi^2\ge 0$ we get

$$|a(\varphi,\phi)| = \left|c\int_\Omega \nabla\varphi\cdot\nabla\phi\,\mathrm{d}x\right| \le c\,\|\nabla\varphi\|_H\,\|\nabla\phi\|_H \le c\,\|\varphi\|_V\,\|\phi\|_V$$

thus a is bounded with constant $\gamma = c > 0$. Furthermore,

$$a(\varphi,\varphi) = c\int_\Omega |\nabla\varphi|^2\ \mathrm{d}x = c\int_\Omega \varphi^2 + |\nabla\varphi|^2 - \varphi^2\,\mathrm{d}x = c\,\|\varphi\|_V^2 - c\,\|\varphi\|_H^2$$

which already means a is coercive with constants $\gamma_1=\gamma_2=c>0$. □

Furthermore, we define the operator $\mathcal{B}\colon U \to L^2(0,T;V')$ such that for all $\varphi \in V$ and a.e. in $[0,T]$ holds

$$\langle (\mathcal{B}u)(t), \varphi \rangle_{V',V} = \sum_{i=1}^{n} u_i^C(t) \int_\Omega \chi_i^C \varphi \,\mathrm{d}x + \sum_{i=1}^{m} u_i^I(y_b - y_a(t)) \int_\Gamma \chi_i^I \varphi \,\mathrm{d}S \quad (2.7)$$

Proposition 2.3. *The operator $\mathcal{B}$ is well-defined, linear and bounded, i.e. there exists a constant $\gamma_3 > 0$ such that for all $u \in U$ holds:*

$$\|\mathcal{B}u\|_{L^2(0,T;V')} \leq \gamma_3 \|u\|_U \quad (2.8)$$

Proof. To show that $\mathcal{B}$ is well-defined we need to show that for arbitrary $u \in U$ and $t \in [0,T]$ a.e. holds $(\mathcal{B}u)(t) \in V'$, i.e. it is a linear and continuous functional mapping from V to $\mathbb{R}$. The linearity of $(\mathcal{B}u)(t)$ follows from the linearity of the integrals, so for $\varphi, \phi \in V$ and $\lambda \in \mathbb{R}$:

$$(\mathcal{B}u)(t)(\lambda\varphi + \phi) = \lambda(\mathcal{B}u)(t)\varphi + (\mathcal{B}u)(t)\phi$$

The continuity of $(\mathcal{B}u)(t)$ follows from Hölder's inequality:

$$\int_\Omega |\chi\varphi| \,\mathrm{d}x = \|\chi\varphi\|_{L^1(\Omega)} \leq \|\chi\|_H \|\varphi\|_H \leq \|\chi\|_H \|\varphi\|_V$$

as $\|\chi\|_H < \infty$ as $\chi = \chi_i^C \in L^\infty(\Omega)$ $(i = 1, \ldots, n)$ and analoguously for the boundary integral. Hence

$$|(\mathcal{B}u)(t)\varphi| \leq C \|\varphi\|_V$$

so $\mathcal{B}$ is well-defined. It is obviously linear in U by its definition:

$$\mathcal{B}(\lambda u + v) = \lambda \mathcal{B}u + \mathcal{B}v$$

It remains to show that $\mathcal{B}$ is bounded, so let $u \in U$ be chosen arbitrarily. Let $\varphi \in V$ then

$$
\begin{aligned}
&\int_0^T \left|\langle (\mathcal{B}u)(t), \varphi\rangle_{V',V}\right|^2 \,\mathrm{d}t \\
&\quad\le \int_0^T \left(\left|\sum_{i=1}^n u_i^C(t) \underbrace{\int_\Omega \chi_i^C \varphi \,\mathrm{d}x}_{=:v_i^C}\right| + \left|\sum_{i=1}^m u_i^I \underbrace{(y_b - y_a(t)) \int_\Gamma \chi_i^I \varphi \,\mathrm{d}S}_{=:v_i^I(t)}\right|\right)^2 \,\mathrm{d}t \\
&\quad= \int_0^T \left(\left|\sum_{i=1}^n u_i^C(t)\, v_i^C\right| + \left|\sum_{i=1}^m u_i^I\, v_i^I(t)\right|\right)^2 \,\mathrm{d}t \\
&\quad\underset{\text{Young}}{\le} 2\int_0^T \left|\sum_{i=1}^n u_i^C(t)\, v_i^C\right|^2 + \left|\sum_{i=1}^m u_i^I\, v_i^I(t)\right|^2 \,\mathrm{d}t \\
&\quad\underset{\text{Schwarz}}{\le} 2\int_0^T \sum_{i=1}^n \left|u_i^C(t)\, v_i^C\right|^2 + \sum_{i=1}^m \left|u_i^I\, v_i^I(t)\right|^2 \,\mathrm{d}t \\
&\quad\le 2 \max_{k=1,\dots,n} \left(v_k^C\right)^2 \int_0^T \sum_{i=1}^n \left(u_i^C(t)\right)^2 \,\mathrm{d}t + 2 \max_{k=1,\dots,m} \int_0^T \left(v_k^I(t)\right)^2 \,\mathrm{d}t \sum_{i=1}^m \left(u_i^I\right)^2 \\
&\quad\le C_1 \left\|u^C\right\|^2_{L^2(0,T;\mathbb{R}^n)} + C_2 \left\|u^I\right\|^2_{\mathbb{R}^m} \\
&\quad\le C \left(\left\|u^C\right\|^2_{L^2(0,T;\mathbb{R}^n)} + \left\|u^I\right\|^2_{\mathbb{R}^m}\right) \;=\; C \left\|u\right\|^2_U
\end{aligned}
$$

From this we conclude that

$$
\begin{aligned}
\left\|\mathcal{B}u\right\|_{L^2(0,T;V')} &= \sup_{\|\varphi\|_V = 1} \left(\int_0^T \left|\langle (\mathcal{B}u)(t), \varphi\rangle_{V',V}\right|^2 \,\mathrm{d}t\right)^{\frac{1}{2}} \\
&\le \sup_{\|\varphi\|_V = 1} C(\varphi) \left\|u\right\|_U \\
&= \gamma_3 \left\|u\right\|_U
\end{aligned}
$$

and thus $\mathcal{B}$ is bounded. □

So, using the bilinear form a (2.5) together with the operator $\mathcal{B}$ (2.7) the PDE (2.4) can be stated weakly as:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \left\langle y(t), \varphi\right\rangle_H + a(y(t), \varphi) &= \left\langle (\mathcal{B}u)(t), \varphi\right\rangle_{V',V} \qquad \forall \varphi \in V \text{ a.e. in } [0,T] \\
y(0) &= y_0 \qquad\qquad \text{in } H
\end{aligned}
\tag{2.9}
$$

where the equality in H means that $\forall \varphi \in H : \left\langle y(0), \varphi\right\rangle_H = \left\langle y_0, \varphi\right\rangle_H$.

Theorem 2.4. *For the symmetric bilinear form $a\colon V \times V \to \mathbb{R}$, $y_0 \in H$ and $\mathcal{B} \in \mathcal{L}(U, L^2(0,T;V'))$ problem (2.9) has a unique weak solution $y \in W(0,T)$ satisfying*

$$\|y\|_{W(0,T)} \leq C\left(\|y_0\|_H + \|u\|_U\right). \tag{2.10}$$

Proof. see for example Section 7.3 in [10]. □

Let us now introduce the Hilbert space $X := W(0,T) \times U$ again endowed with the natural product topology, i.e. for $x = (y,u) \in X$ we have the induced norm $\|x\|_X = (\|y\|^2_{W(0,T)} + \|u\|^2_U)^{1/2}$.

We infer that $X^{int}_{ad} = W(0,T) \times U^{int}_{ad}$ is non-empty because there exists a unique weak solution $y \in W(0,T)$ for all $u \in U$ according to Theorem 2.4 and U^{int}_{ad} is a non-empty subset of U.

Hence, we can state the problem

$$\min J(\mathcal{E}y, u) \quad \text{s.t.} \begin{cases} (y,u) \in X^{int}_{ad} \\ y \text{ solves } (2.9) \end{cases} \tag{P}$$

with the canonical embedding $\mathcal{E}\colon W(0,T) \to L^2(0,T;H)$ which is linear and bounded and maps every function $y \in W(0,T)$ onto the same function in $L^2(0,T;H)$.

Furthermore, a solution y can be split into two parts, one depending on the fixed initial condition y_0 and the other depending linearly on the control variable u. So let $\hat{y} \in W(0,T)$ be the unique weak solution to

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle \hat{y}(t), \varphi\rangle_H + a(\hat{y}(t), \varphi) = 0 \qquad & \forall \varphi \in V \text{ a.e. in } [0,T] \\ \hat{y}(0) = y_0 \qquad & \text{in } H \end{aligned} \tag{2.11}$$

i.e. (2.9) with $u = 0$. By Theorem 2.4 we know that (2.9) admits a unique weak solution for all controls $u \in U$ and any given initial value $y_0 \in H$. Thus we can define the linear and by (2.10) bounded solution operator $\mathcal{S}\colon U \to W(0,T), u \mapsto \mathcal{S}u = y$ with y being the unique solution to

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle y(t), \varphi\rangle_H + a(y(t), \varphi) = \langle (\mathcal{B}u)(t), \varphi\rangle_{V',V} \quad & \forall \varphi \in V \text{ a.e. in } [0,T] \\ y(0) = 0 \qquad & \text{in } H. \end{aligned} \tag{2.12}$$

i.e. with homogeneous initial condition $y_0 = 0$. This means, the solution y to (2.9) is a dependent variable and given as $y = \hat{y} + \mathcal{S}u$. Consequently, we can introduce the reduced cost functional

$$\hat{J}\colon U \to \mathbb{R},\ \hat{J}(u) = J(\mathcal{E}(\hat{y} + \mathcal{S}u), u)$$

and consider the reduced optimal control problem:

$$\min \hat{J}(u) \text{ s.t. } u \in U_{ad}^{int} \tag{$\hat{\mathsf{P}}$}$$

Clearly, if $\bar{u}$ is the optimal solution to ($\hat{\mathsf{P}}$), $\bar{x} = (\mathcal{E}(\hat{y} + \mathcal{S}\bar{u}), \bar{u})$ is the optimal solution to (P). And if $\bar{x} = (\bar{y}, \bar{u})$ solves (P), then $\bar{u}$ is the optimal solution to ($\hat{\mathsf{P}}$).

2.2 Existence of a weak optimal solution

Note that $J \to \infty$ for $\|u\|_U \to \infty$, so we could also in an unrestricted case consider the admissible set bounded with suitably small $u_a \in U$ and large $u_b \in U$ such that $-\infty < u_a \leq u_b < \infty$. As

$$U_{ad}^{int} = \{(u^C, u^I) \in U \mid u^C \in [u_a^C, u_b^C], u^I \in [u_a^I, u_b^I] \cap \mathbb{Z}^m\}$$

where $[u_a^I, u_b^I] \cap \mathbb{Z}^m$ is obviously finite, we can observe that

$$\min_{u \in U_{ad}^{int}} \hat{J}(u) = \min_{u^I \in [u_a^I, u_b^I] \cap \mathbb{Z}^m} \underbrace{\left(\min_{u^C \in [u_a^C, u_b^C]} \{\hat{J}(u) \mid u = (u^C, u^I)\} \right)}_{=:(\hat{Q}_{u^I})}$$

Proposition 2.5. *For any given $u^I \in [u_a^I, u_b^I] \cap \mathbb{Z}^m$ problem $(\hat{Q}_{u^I})$ admits a unique solution $u^C \in [u_a^C, u_b^C]$.*

Proof. Note that $[u_a^C, u_b^C]$ is a non-empty, bounded, closed and convex subset of $L^2(0, T; \mathbb{R}^n)$ and

$$(\hat{Q}_{u^I}) \iff \min_{u^C \in [u_a^C, u_b^C]} \frac{\alpha_Q}{2} \|\mathcal{E}\mathcal{S}u - y_d\|^2_{L^2(0,T;H)} + \frac{1}{2} \left\|u^C\right\|^2_{\alpha^C}$$

where

$$\|\cdot\|_{\alpha^C} := \left(\sum_{i=1}^{n} \int_0^T (\sqrt{\alpha_i^C} u_i^C(t))^2 \, \mathrm{d}t \right)^{\frac{1}{2}}$$

defines a weighted norm on the Hilbert space $L^2(0,T;\mathbb{R}^n)$ as $\alpha_i^C > 0$ for $i = 1,\dots,n$. So the claim follows from Theorem 2.14 in [10]. □

Hence, there exists a solution to $(\hat{\mathsf{P}})$

$$\bar{u} = \underset{u^I \in [u_a^I, u_b^I] \cap \mathbb{Z}^m}{\operatorname{argmin}} \{\hat{J}(u^C(u^I), u^I)\}$$

since the integer set $[u_a^I, u_b^I] \cap \mathbb{Z}^m$ is finite.

Unfortunately, we cannot provide uniqueness of an optimal solution $\bar{u}$. It might happen that different controls yield the same objective. One might think of a cheap insulation alongside more heating and less heating alongside a better insulation giving similar temperatures. Secondly, small temperature differences and increased costs might result in similar values in $\hat{J}$ as a large deviation from the desired temperature alongside small costs.

Definition 2.6. *The relaxed optimal control problem or relaxation to* $(\hat{\mathsf{P}})$ *is given by dropping the integrality constraint for the admissible controls*

$$\min \hat{J}(u) \text{ s.t. } u \in U_{ad}(a,b) \qquad (\hat{\mathsf{R}}_{ab})$$

where $U_{ad}(a,b) = \{u \in U \mid a \le u^I \le b\}$ *for given* $a, b \in \mathbb{Z}^m$ *with* $a \le b$.

Remark 4. The relaxed problem $(\hat{\mathsf{R}}_{ab})$ is uniquely solvable as $\hat{J}$ is strictly convex. In the branch-and-bound algorithm which we will present in Chapter 4 in order to solve $(\hat{\mathsf{P}})$ we will have to solve many of these relaxed problems $(\hat{\mathsf{R}}_{ab})$ with varying lower and upper bounds $a, b \in \mathbb{Z}^m$. Consequently, we incorporated them in the notation of the admissible set $U_{ad}(a,b)$. ◊

Remark 5. Theorem 2.14 in [10] cannot provide a unique solution neither to $(\hat{\mathsf{P}})$ because U_{ad}^{int} is not convex (nor to relaxed problem $(\hat{\mathsf{R}}_{ab})$ with $a = u_a^I$ and $b = u_b^I$ because including the subtraction of the reference control ω does not define a norm on U anymore).
We will keep the possibility of several optimal solutions of $(\hat{\mathsf{P}})$ in our minds when it comes to the numerical part. Otherwise, multiobjective methods might be an idea how to manage those contradictory goals combined in $\hat{J}$. ◊

2.3 First-order necessary optimality conditions

Proposition 2.7. *A control $\bar{\mathfrak{u}}_{ab} \in U_{ad}(a,b)$ is an optimal solution to $(\hat{\mathsf{R}}_{ab})$, if and only if it satisfies the variational inequality*

$$\left\langle \nabla \hat{J}(\bar{\mathfrak{u}}_{ab}), u - \bar{\mathfrak{u}}_{ab} \right\rangle_U \geq 0 \tag{2.13}$$

for all $u \in U_{ad}(a,b)$.

Proof. As $U_{ad}(a,b)$ is a convex subset of the Hilbert space U and since $\hat{J}$ is convex and Fréchet-differentiable, the statement follows from Lemma 2.21 in [10]. □

Recall that $y_d \in L^2(0,T;H)$ and $y = \hat{y} + \mathcal{S}u \in W(0,T)$. Let us define $\mathcal{G} := \mathcal{E}\mathcal{S} \colon U \to L^2(0,T;H)$ and $z_d := y_d - \mathcal{E}\hat{y} \in L^2(0,T;H)$. Let further

$$\begin{aligned} F &\colon U \to \mathbb{R} \\ u = (u^C, u^I) &\mapsto \frac{1}{2} \sum_{i=1}^{n} \alpha_i^C \int_0^T u_i^C(t)^2 \, \mathrm{d}t + \frac{1}{2} \sum_{i=1}^{m} \alpha_i^I (u_i^I - \omega_i)^2 \end{aligned}$$

Then the cost functional reformulates to

$$\hat{J}(u) = \frac{\alpha_Q}{2} \left\| \mathcal{G}u - z_d \right\|^2_{L^2(0,T;H)} + F(u) \tag{2.14}$$

which is Fréchet-differentiable. For the second summand we immediately find

$$\nabla F(u) = \begin{pmatrix} \alpha_1^C u_1^C(\cdot) \\ \vdots \\ \alpha_n^C u_n^C(\cdot) \\ \hline \alpha_1^I (u_1^I - \omega_1) \\ \vdots \\ \alpha_m^I (u_m^I - \omega_m) \end{pmatrix}$$

so now we focus on the first summand of (2.14) in (2.13). By the chain-rule for Fréchet-derivatives, Theorem 2.20 in [10], we achieve

$$\frac{\alpha_Q}{2} \left\langle \nabla(\left\| \mathcal{G}\bar{\mathfrak{u}}_{ab} - z_d \right\|^2_{L^2(0,T;H)}), u - \bar{\mathfrak{u}}_{ab} \right\rangle_U = \alpha_Q \left\langle \mathcal{G}\bar{\mathfrak{u}}_{ab} - z_d, \mathcal{G}(u - \bar{\mathfrak{u}}_{ab}) \right\rangle_{L^2(0,T;H)}$$

Recall that $\mathcal{S}\colon U \to W(0,T)$ is the solution operator to the weak formulation of the PDE with homogeneous initial condition. We define its adjoint operator

$$\begin{aligned} \mathcal{S}'\colon W(0,T)' &\to U' \sim U \\ \varphi &\mapsto \mathcal{S}'\varphi \end{aligned}$$

such that for all $u \in U$ holds

$$\langle \mathcal{S}'\varphi, u\rangle_U = \langle \varphi, \mathcal{S}u\rangle_{W(0,T)',W(0,T)}$$

with a dual pairing on the right-hand side. By the Riesz representation theorem (see Theorem 2.9 in [10]) it is well-defined and we identify the Hilbert space U with its dual U' by the Riesz isomorphism.

Recall that $\mathcal{E}\colon W(0,T) \to L^2(0,T;H), \varphi \mapsto \varphi$ is the canonical embedding. Also here we define the adjoint operator

$$\begin{aligned} \mathcal{E}'\colon L^2(0,T;H) &\to W(0,T)' \\ \varphi &\mapsto (y \mapsto \langle \varphi, \mathcal{E}y\rangle_{L^2(0,T;H)}) \end{aligned}$$

which satisfies

$$\langle \mathcal{E}'\varphi, y\rangle_{W(0,T)',W(0,T)} = \langle \varphi, \mathcal{E}y\rangle_{L^2(0,T;H)}$$

for all $y \in W(0,T)$ and all $\varphi \in L^2(0,T;H)$.

Note that in both cases we do not identify the Hilbert space $W(0,T)$ with its dual as this would require the use of some unwanted $W(0,T)$-scalar products.

Using the adjoints the upper reformulates to

$$\begin{aligned} &\alpha_Q \left\langle \mathcal{G}\bar{\mathfrak{u}}_{ab} - z_d, \mathcal{G}(u - \bar{\mathfrak{u}}_{ab})\right\rangle_{L^2(0,T;H)} \\ &= \alpha_Q \left\langle \mathcal{E}\mathcal{S}\bar{\mathfrak{u}}_{ab} - z_d, \mathcal{E}\mathcal{S}(u - \bar{\mathfrak{u}}_{ab})\right\rangle_{L^2(0,T;H)} \\ &= \alpha_Q \left\langle \mathcal{E}'(\mathcal{E}\mathcal{S}\bar{\mathfrak{u}}_{ab} - z_d), \mathcal{S}(u - \bar{\mathfrak{u}}_{ab})\right\rangle_{W(0,T)',W(0,T)} \end{aligned}$$

Analoguously to Section 4.3 in [5] we introduce the two linear and bounded operators

$$\begin{aligned} \Theta\colon W(0,T) &\to W(0,T)' \\ y &\mapsto \alpha_Q\, \mathcal{E}'\mathcal{E}y \end{aligned}$$

and

$$\begin{aligned} \Xi \colon L^2(0,T;H) &\to W(0,T)' \\ z &\mapsto \alpha_Q \, \mathcal{E}' z \end{aligned}$$

which translate to

$$\begin{aligned} \langle \Theta y, \phi \rangle_{W(0,T)',W(0,T)} &= \int_0^T \alpha_Q \, \langle (\mathcal{E}y)(t), (\mathcal{E}\phi)(t) \rangle_H \, \mathrm{d}t \\ &= \int_0^T \alpha_Q \, \langle y(t), \phi(t) \rangle_H \, \mathrm{d}t \end{aligned}$$

and

$$\begin{aligned} \langle \Xi z, \phi \rangle_{W(0,T)',W(0,T)} &= \int_0^T \alpha_Q \, \langle z(t), (\mathcal{E}\phi)(t) \rangle_H \, \mathrm{d}t \\ &= \int_0^T \alpha_Q \, \langle z(t), \phi(t) \rangle_H \, \mathrm{d}t \end{aligned}$$

Note that we relinquish using $\mathcal{E}_V \colon V \to H, \varphi \mapsto \varphi$ in order to not overdose on embedding operators, so by $\langle z(t), \phi(t) \rangle_H$ we mean $\langle \mathcal{E}_V(z(t)), \mathcal{E}_V(\phi(t)) \rangle_H$. Plugging this into above and using the adjoint solution operator yields:

$$\begin{aligned} & \alpha_Q \, \langle \mathcal{E}'(\mathcal{E}\mathcal{S}\bar{\mathfrak{u}}_{ab} - z_d), \mathcal{S}(u - \bar{\mathfrak{u}}_{ab}) \rangle_{W(0,T)',W(0,T)} \\ = \; & \langle \Theta \mathcal{S}\bar{\mathfrak{u}}_{ab} - \Xi z_d, \mathcal{S}(u - \bar{\mathfrak{u}}_{ab}) \rangle_{W(0,T)',W(0,T)} \\ = \; & \langle \mathcal{S}'(\Theta \mathcal{S}\bar{\mathfrak{u}}_{ab} - \Xi z_d), u - \bar{\mathfrak{u}}_{ab} \rangle_U \end{aligned}$$

Consequently, in order to compute the derivative of the cost functional we need to compute the adjoint. Section 2.10 in [10] gives a profound explanation how to determine the adjoint equation.

In our setting the adjoint variable p is given as the for all $u \in U$ unique solution of

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t} \langle p(t), \varphi \rangle_H + a(p(t), \varphi) &= \alpha_Q \, \langle y_d(t) - y(t), \varphi \rangle_H \\ p(T) &= 0 \end{aligned} \tag{2.15}$$

for all $\varphi \in V$ a.e. in $[0,T]$. Using $y = \hat{y} + \mathcal{S}u$ we can also split the adjoint into $p = \hat{p} + \mathcal{A}u$ where control-independent part $\hat{p}$ is the solution to

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t} \langle \hat{p}(t), \varphi \rangle_H + a(\hat{p}(t), \varphi) &= \alpha_Q \, \langle y_d(t) - \hat{y}(t), \varphi \rangle_H \\ \hat{p}(T) &= 0 \end{aligned} \tag{2.16}$$

for all $\varphi \in V$ a.e. in $[0,T]$ and $\mathcal{A}\colon U \to W(0,T), u \mapsto \mathcal{A}u = p$ is the well-defined, linear and bounded solution operator to

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}\left\langle p(t),\varphi\right\rangle_H + a(p(t),\varphi) &= -\alpha_Q\left\langle (\mathcal{S}u)(t),\varphi\right\rangle_H \\ p(T) &= 0 \end{aligned} \tag{2.17}$$

for all $\varphi \in V$ a.e. in $[0,T]$.

We state the following two lemma along Lemma 4.9 and 4.10 in [5] to show that $-\mathcal{S}'\Theta\mathcal{S} = \mathcal{B}'\mathcal{A} \in \mathcal{L}(U)$ and $\mathcal{S}'\Xi(y_d - \mathcal{E}\hat{y}) = \mathcal{B}'\hat{p}$ which we will use to again reformulate the first summand in the variational inequality (2.13).

Lemma 2.8. *Let $u, v \in U$ be chosen arbitrarily. We set $y = \mathcal{S}u \in W(0,T)$ and $p = \mathcal{A}v \in W(0,T)$. Then*

$$\int_0^T \left\langle (\mathcal{B}u)(t), p(t)\right\rangle_{V',V} \,\mathrm{d}t = -\alpha_Q \int_0^T \left\langle (\mathcal{S}v)(t), y(t)\right\rangle_H \,\mathrm{d}t.$$

Proof. It holds

$$\begin{aligned} &\int_0^T \left\langle (\mathcal{B}u)(t), p(t)\right\rangle_{V',V} \,\mathrm{d}t = \int_0^T \left\langle y_t(t), p(t)\right\rangle_{V',V} + a(y(t),p(t)) \,\mathrm{d}t \\ &= \int_0^T -\left\langle p_t(t), y(t)\right\rangle_{V',V} + a(p(t),y(t)) \,\mathrm{d}t + \langle \underbrace{p(T)}_{=0}, y(T)\rangle_H - \langle p(0), \underbrace{y(0)}_{=0}\rangle_H \\ &= -\alpha_Q \int_0^T \left\langle (\mathcal{S}v)(t), y(t)\right\rangle_H \,\mathrm{d}t \end{aligned}$$

where the first equality is derived from the state equation (2.12), the second from integration by parts and and the third from the adjoint equation (2.17). □

Lemma 2.9. *For the previously defined operators holds $\mathcal{B}'\mathcal{A} = -\mathcal{S}'\Theta\mathcal{S} \in \mathcal{L}(U)$ and $\mathcal{B}'\hat{p} = \mathcal{S}'\Xi(y_d - \mathcal{E}\hat{y})$, where $\hat{p}$ is the solution to* (2.16) *and $\mathcal{B}'\colon L^2(0,T,V) \mapsto U$ is the adjoint operator to $\mathcal{B}$.*

Proof. Let $u, v \in U$ be arbitrary. We set $y = \mathcal{S}u \in W(0,T), p = \mathcal{A}v \in W(0,T) \subset L^2(0,T;V)$. Recall that we identify U with its dual space U'.

By definition of Θ and from Lemma 2.8 we infer that

$$\begin{aligned}\langle \mathcal{S}'\Theta\mathcal{S}v, u\rangle_U &= \langle \Theta\mathcal{S}v, \mathcal{S}u\rangle_{W(0,T)',W(0,T)} = \alpha_Q \langle \mathcal{E}\mathcal{S}v, \mathcal{E}\mathcal{S}u\rangle_{L^2(0,T;H)} \\ &= \alpha_Q \int_0^T \langle (\mathcal{S}v)(t), y(t)\rangle_H \, \mathrm{d}t \\ &= -\langle \mathcal{B}u, p\rangle_{L^2(0,T;V'),L^2(0,T;V)} = -\langle u, \mathcal{B}'p\rangle_U = -\langle \mathcal{B}'\mathcal{A}v, u\rangle_U\end{aligned}$$

And by definition of Ξ and from integration by parts follows

$$\begin{aligned}\langle \mathcal{S}'\Xi(y_d - \mathcal{E}\hat{y}), u\rangle_U &= \langle \Xi(y_d - \mathcal{E}\hat{y}), \mathcal{S}u\rangle_{W(0,T)',W(0,T)} \\ &= \int_0^T \alpha_Q \langle y_d(t) - \hat{y}(t), y(t)\rangle_H \, \mathrm{d}t \\ &= \int_0^T -\langle \hat{p}_t(t), y(t)\rangle_{V',V} + a(\hat{p}(t), y(t)) \, \mathrm{d}t \\ &= \int_0^T \langle y_t(t), \hat{p}(t)\rangle_{V',V} + a(y(t), \hat{p}(t)) \, \mathrm{d}t \\ &= \int_0^T \langle (\mathcal{B}u)(t), \hat{p}(t)\rangle_{V',V} \, \mathrm{d}t \\ &= \langle \mathcal{B}u, \hat{p}\rangle_{L^2(0,T;V'),L^2(0,T;V)} = \langle u, \mathcal{B}'\hat{p}\rangle_U \qquad \square\end{aligned}$$

Using this we achieve

$$\langle \mathcal{S}'(\Theta\mathcal{S}\bar{\mathfrak{u}}_{ab} - \Xi z_d), u - \bar{\mathfrak{u}}_{ab}\rangle_U = -\langle \mathcal{B}' \underbrace{(\mathcal{A}\bar{\mathfrak{u}}_{ab} + \hat{p})}_{=:\bar{p}_{ab}}, u - \bar{\mathfrak{u}}_{ab}\rangle_U$$

so altogether, we reformulated the variational inequality (2.13) as:

$$\left\langle \nabla \hat{J}(\bar{\mathfrak{u}}_{ab}), u - \bar{\mathfrak{u}}_{ab}\right\rangle_U = \langle \nabla F(\bar{\mathfrak{u}}_{ab}) - \mathcal{B}'\bar{p}_{ab}, u - \bar{\mathfrak{u}}_{ab}\rangle_U \tag{2.18}$$

where $\bar{p}_{ab} = \hat{p} + \mathcal{A}\bar{\mathfrak{u}}_{ab}$. This leads to the the first-order necessary and (due to the convexity of $\hat{J}$ sufficient) optimality conditions in the following theorem.

Theorem 2.10. *If and only if $\bar{\mathfrak{u}}_{ab}$ satisfies together with the state variable $\bar{y}_{ab}$ and the adjoint variable $\bar{p}_{ab}$ the first-order optimality system*

$$\begin{aligned}&\bar{y}_{ab} = \hat{y} + \mathcal{S}\bar{\mathfrak{u}}_{ab}, \quad \bar{p}_{ab} = \hat{p} + \mathcal{A}\bar{\mathfrak{u}}_{ab}, \quad u_a \le \bar{\mathfrak{u}}_{ab} \le u_b, \\ &\langle \nabla F(\bar{\mathfrak{u}}_{ab}) - \mathcal{B}'\bar{p}_{ab}, u - \bar{\mathfrak{u}}_{ab}\rangle_U \ge 0 \quad \text{for all } u \in U_{ad}(a,b)\end{aligned} \tag{2.19}$$

then $\bar{\mathfrak{u}}_{ab}$ is an optimal solution to $(\hat{\mathsf{R}}_{ab})$.

Proof. By Proposition 2.7, by definition of the state and adjoint variables and operators and the computation of $\nabla \hat{J}(u) = \nabla F(u) - \mathcal{B}'p$. □

2.4 Finite element discretization

To numerically tackle the problem we have to discretize as well time and space. The time interval shall be approximated by a – for simplicity – equidistant time grid $0 = t_1 < t_2 < \ldots < t_N = T$ and the spatial discretization shall be realized by the finite element (FE) method which we quickly recall.

Given a triangulation of the domain Ω with vertices $v_1, \ldots, v_r$ we define piecewise linear ansatz functions $\varphi_1, \ldots, \varphi_r$ satisfying $\varphi_i(v_j) = \delta_{ij}$ and spanning the subspace $V_h \subset V$. The Kronecker delta is defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

and the number of nodes r gives the degrees of freedom.
To the state y and the adjoint p we want to find approximate solutions y^h and p^h in the r-dimensional space V_h by using the Galerkin ansatz:

$$y^h(t,x) = \sum_{i=1}^{r} \mathbf{y}_i^h(t)\, \varphi_i(x) \quad \text{and} \quad p^h(t,x) = \sum_{i=1}^{r} \mathbf{p}_i^h(t)\, \varphi_i(x) \tag{2.20}$$

with coefficient functions $\mathbf{y}^h, \mathbf{p}^h \colon [0,T] \to \mathbb{R}^r$. Plugging this ansatz into the weak formulation of the state equation (2.9) and the adjoint equation (2.15) yields for all $\varphi \in V_h$ a.e. in $[0,T]$:

$$\begin{aligned} \sum_{i=1}^{r} \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{y}_i^h(t) \langle \varphi_i, \varphi \rangle_H + \sum_{i=1}^{r} \mathbf{y}_i^h(t)\, a(\varphi_i, \varphi) &= \langle (\mathcal{B}u)(t), \varphi \rangle_{V',V} \\ \sum_{i=1}^{r} \mathbf{y}_i^h(0) \langle \varphi_i, \varphi \rangle_H &= \langle y_0, \varphi \rangle_H \end{aligned} \tag{2.21}$$

and

$$\begin{aligned} -\sum_{i=1}^{r} \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{p}_i^h(t) \langle \varphi_i, \varphi \rangle_H + \sum_{i=1}^{r} \mathbf{p}_i^h(t)\, a(\varphi_i, \varphi) &= \alpha_Q \left\langle y_d(t) - y^h(t), \varphi \right\rangle_H \\ \sum_{i=1}^{r} \mathbf{p}_i^h(T) \langle \varphi_i, \varphi \rangle_H &= 0 \end{aligned} \tag{2.22}$$

Choosing the ansatz functions as well as test functions $\varphi = \varphi_i$ for $i = 1, \dots, r$ and defining the so-called

- mass matrix $M = (M_{ij})_{1 \le i,j \le r}$ where $M_{ij} = \langle \varphi_i, \varphi_j \rangle_H$
- stiffness matrix $A = (A_{ij})_{1 \le i,j \le r}$ where $A_{ij} = a(\varphi_i, \varphi_j)$
- right-hand side $B(u,t) = (B_i(u,t))_{1 \le i \le r}$ with $B_i(u,t) = \langle (\mathcal{B}u)(t), \varphi_i \rangle_{V',V}$
- the projection $y_0^h = (\langle y_0, \varphi_i \rangle_H)_{i=1,\dots,r}$ of the initial condition $y_0 \in H$ onto the FE basis
- the projection $y_d^h(t) = (\langle y_d(t), \varphi_i \rangle_H)_{i=1,\dots,r}$ of the desired temperature $y_d(t) \in H$ onto the FE basis

we get the ODE systems

$$\begin{aligned} M\,\dot{\mathbf{y}}^h(t) + A\,\mathbf{y}^h(t) &= B(u,t) \\ M\,\mathbf{y}^h(0) &= y_0^h \end{aligned} \tag{2.23}$$

and

$$\begin{aligned} -M\,\dot{\mathbf{p}}^h(t) + A\,\mathbf{p}^h(t) &= \alpha_Q \left(y_d^h(t) - M\,\mathbf{y}^h \right) \\ \mathbf{p}^h(T) &= 0 \end{aligned} \tag{2.24}$$

which can be solved by a numerical time-stepping algorithm, for example the implicit Euler method, Crank-Nicolson method or Rannacher Smoothing. The latter does four implicit Euler steps with halved step size followed by regular Crank-Nicolson steps and thus combines the advantages of the two methods – robustness to discontinuities of the initial condition and convergence order 2.

Again we split $y^h = \hat{y}^h + \mathcal{S}^h u$ and $p^h = \hat{p}^h + \mathcal{A}^h u$ with the control-independent parts $\hat{y}^h$ and $\hat{p}^h$ to avoid their repeated computation during the optimization process when we attempt to find an optimal control. Analoguously as in Section 2.1 resp. 2.3 we define the operators $\mathcal{S}^h, \mathcal{A}^h \colon U \to W(0,T)$ as the solutions to the homogeneous state ODE (2.23) with $y_0^h = 0$ and homogeneous adjoint ODE (2.24) with $\hat{\mathbf{y}}^h = 0$. The same properties as for the operators $\mathcal{S}$ and $\mathcal{A}$ also hold for these, $\mathcal{S}^h$ and $\mathcal{A}^h$ are linear and bounded and satisfy

$$-(\mathcal{S}^h)'\Theta\mathcal{S}^h = \mathcal{B}'\mathcal{A}^h \in \mathcal{L}(U) \quad \text{and} \quad (\mathcal{S}^h)'\Xi(y_d - \mathcal{E}\hat{y}^h) = \mathcal{B}'\hat{p}^h$$

which follows analoguously to Lemmas 2.8 and 2.9.

Now we will replace $(\hat{\mathsf{P}})$ by discretized problem

$$\min \hat{J}^h(u) := \min J(\mathcal{E}(\hat{y}^h + \mathcal{S}^h u), u) \quad \text{s.t. } u \in U_{ad}^{int} \tag{$\hat{P}^h$}$$

and the relaxation $(\hat{\mathsf{R}}_{ab})$ by

$$\min \hat{J}^h(u) \text{ s.t. } u \in U_{ad}(a,b) \tag{$\hat{R}^h_{ab}$}$$

As already said in Remark 4 we have to solve many of these for varying $a, b \in \mathbb{Z}^m$ during the optimization process in the branch-and-bound algorithm.

By the same arguments as in Proposition 2.7 a control $\bar{u}_{ab} \in U_{ad}(a,b)$ is an optimal solution to $(\hat{R}^h_{ab})$, iff it satisfies the variational inequality

$$\left\langle \nabla \hat{J}^h(\bar{u}_{ab}), u - \bar{u}_{ab} \right\rangle_U \geq 0 \quad \text{for all } u \in U_{ad}(a,b) \tag{2.25}$$

which can – analoguously to (2.19) – be reformulated into the first-order necessary optimality system

$$\begin{aligned} &\bar{y}^h_{ab} = \hat{y}^h + \mathcal{S}^h \bar{u}_{ab}, \quad \bar{p}^h_{ab} = \hat{p}^h + \mathcal{A}^h \bar{u}_{ab}, \quad \bar{u}_{ab} \in U_{ad}(a,b), \\ &\left\langle \nabla F(\bar{u}_{ab}) - \mathcal{B}' \bar{p}^h_{ab}, u - \bar{u}_{ab} \right\rangle_U \geq 0 \quad \text{for all } u \in U_{ad}(a,b) \end{aligned} \tag{2.26}$$

because

$$\begin{aligned} \nabla \hat{J}^h(u) &= \nabla F(u) + (\mathcal{S}^h)'(\Theta \mathcal{S}^h u - \Xi z_d) \\ &= \nabla F(u) - \mathcal{B}'(\mathcal{A}^h u + \hat{p}^h) \end{aligned} \tag{2.27}$$

as in Section 2.3.

2.5 A posteriori error estimates

Note that (2.25) only holds for an optimal control $\bar{u}_{ab}$, but we will derive a similar inequality for an arbitrary control in $U_{ad}(a,b)$.

Lemma 2.11. *The cost functional $\hat{J}^h$ is twice Fréchet-differentiable with constant second derivative*

$$(\nabla^2 \hat{J}^h)v = (\nabla^2 F)v + (\mathcal{S}^h)' \Theta \mathcal{S}^h v, \quad v \in U \tag{2.28}$$

and uniformely coercive

$$\left\langle (\nabla^2 \hat{J}^h)v, v \right\rangle_U \geq \bar{\alpha} \|v\|_U^2 \tag{2.29}$$

with coercivity constant $\bar{\alpha} > 0$.

Proof. The second derivative follows directly from (2.27) and its uniform coercitivity can be seen with the help of the operators $(\mathcal{S}^h)'$ and Θ as follows:

$$\begin{aligned}
\left\langle (\nabla^2 \hat{J}^h)v, v \right\rangle_U &= \left\langle (\nabla^2 F)v, v \right\rangle_U + \left\langle (\mathcal{S}^h)'\Theta \mathcal{S}^h v, v \right\rangle_U \\
&= \sum_{i=1}^{n} \alpha_i^C \int_0^T v_i^C(t)^2 \, \mathrm{d}t + \sum_{i=1}^{m} \alpha_i^I (v_i^I)^2 + \alpha_Q \int_0^T \left\langle (\mathcal{S}^h v)(t), (\mathcal{S}^h v)(t) \right\rangle_H \, \mathrm{d}t \\
&\geq \bar{\alpha} \|v\|_U^2 + \underbrace{\alpha_Q \int_0^T \left\| (\mathcal{S}^h v)(t) \right\|_H^2 \, \mathrm{d}t}_{\geq 0} \\
&\geq \bar{\alpha} \|v\|_U^2
\end{aligned}$$

for all $v \in U$ with coercitivity constant $\bar{\alpha} = \min\{\alpha_1^C, \ldots, \alpha_n^C, \alpha_1^I, \ldots, \alpha_m^I\}$. □

Let $\tilde{u} \in U_{ad}(a, b)$ be an arbitrarily chosen control from the admissible set such that $\tilde{u} \neq \bar{u}_{ab}$. Then $\tilde{u}$ does not satisfy the variational inequality (2.25), but there exists a perturbation function $\xi = (\xi^C, \xi^I) \in U$ (cf. Section 3 in [11]) such that

$$\left\langle \nabla \hat{J}^h(\tilde{u}) + \xi, u - \tilde{u} \right\rangle_U \geq 0 \quad \text{for all } u \in U_{ad}(a, b) \tag{2.30}$$

In the following theorem we will define such $\xi \in U$ and state an a posteriori estimate for $\|\tilde{u} - \bar{u}_{ab}\|_U$.

Theorem 2.12. *Let* $\tilde{u} = (\tilde{u}^C, \tilde{u}^I) \in U_{ad}(a, b)$ *satisfying* $\tilde{u} \neq \bar{u}_{ab}$. *We define* $\xi(\tilde{u}) = (\xi^C(\tilde{u}), \xi^I(\tilde{u})) \in U$ *by*

$$\xi_j^C(\tilde{u})(t) = \begin{cases} \max\{0, -(\nabla^C \hat{J}^h(\tilde{u}))_j(t)\} & \text{if } \tilde{u}_j^C(t) = (u_a^C)_j(t) \\ \min\{0, -(\nabla^C \hat{J}^h(\tilde{u}))_j(t)\} & \text{if } \tilde{u}_j^C(t) = (u_b^C)_j(t) \\ -(\nabla^C \hat{J}^h(\tilde{u}))_j(t) & \text{otherwise} \end{cases} \tag{2.31a}$$

for $j = 1, \ldots, n$ *and*

$$\xi_j^I(\tilde{u}) = \begin{cases} 0 & \text{if } a_j = b_j \\ \max\{0, -(\nabla^I \hat{J}^h(\tilde{u}))_j\} & \text{if } \tilde{u}_j^I = a_j \\ \min\{0, -(\nabla^I \hat{J}^h(\tilde{u})_j\} & \text{if } \tilde{u}_j^I = b_j \\ -(\nabla^I \hat{J}^h(\tilde{u}))_j & \text{otherwise} \end{cases} \tag{2.31b}$$

for $j = 1, \ldots, m$. *Then,* $\xi(\tilde{u})$ *satisfies the perturbed variational inequality* (2.30) *and the a posteriori error estimator*

$$\|\tilde{u} - \bar{u}_{ab}\|_U \leq \Delta_{\text{apost}}(\tilde{u}) = \bar{\alpha}^{-1} \|\xi(\tilde{u})\|_U \tag{2.32}$$

holds with the coercivity constant $\bar{\alpha}$ *from* (2.29).

Proof. By definition of ξ holds for all $v \in U_{ad}(a,b)$

$$\left\langle \nabla \hat{J}^h(\tilde{u}) + \xi(\tilde{u}), v - \tilde{u} \right\rangle_U \geq 0$$

as all summands in the inner product are non-negative, as for inactive indices (i.e. no constraint active, both *otherwise* cases in the definition of ξ) holds

$$(\nabla \hat{J}^h(\tilde{u}) + \xi(\tilde{u}))_j = 0$$

and for active indices the definition of ξ guarantees that the same sign of both factors

$$\operatorname{sgn}((\nabla \hat{J}^h(\tilde{u}) + \xi(\tilde{u}))_j) = \operatorname{sgn}((v - \tilde{u})_j)$$

making their product always non-negative. Now choose $v = \bar{u}_{ab}$ and add it to the variational inequality (2.25) where we chose $u = \tilde{u}$ to obtain

$$\begin{aligned} 0 &\leq \left\langle \nabla \hat{J}^h(\bar{u}_{ab}), \tilde{u} - \bar{u}_{ab} \right\rangle_U + \left\langle \nabla \hat{J}^h(\tilde{u}) + \xi(\tilde{u}), \bar{u}_{ab} - \tilde{u} \right\rangle_U \\ &= -\left\langle \nabla \hat{J}^h(\bar{u}_{ab}) - \nabla \hat{J}^h(\tilde{u}), \bar{u}_{ab} - \tilde{u} \right\rangle_U + \langle \xi(\tilde{u}), \bar{u}_{ab} - \tilde{u} \rangle_U \end{aligned}$$

Applying the mean value theorem and exploiting that the second derivative is constant yields

$$\begin{aligned} 0 &\leq -\left\langle (\nabla^2 \hat{J}^h)(\bar{u}_{ab} - \tilde{u}), \bar{u}_{ab} - \tilde{u} \right\rangle_U + |\langle \xi(\tilde{u}), \bar{u}_{ab} - \tilde{u} \rangle_U| \\ &\leq -\bar{\alpha} \left\| \tilde{u} - \bar{u}_{ab} \right\|_U^2 + \left\| \xi(\tilde{u}) \right\|_U \left\| \tilde{u} - \bar{u}_{ab} \right\|_U \end{aligned}$$

where the latter inequality follows by the coercitivity from (2.29) and the Cauchy-Schwarz inequality. Division by $\|\tilde{u} - \bar{u}_{ab}\|_U > 0$ and rearranging gives the result (2.32). □

To sum it up, solving $(\hat{P}^h)$ requires solving several relaxed problems $(\hat{R}^h_{ab})$ which each mean solving an optimality system of the type (2.26) which requires many state and adjoint solutions. In order to speed up the computation, we want to approximate the state and adjoint equation by reduced-order models (ROM) which will be computed with the help of the proper orthogonal decomposition (POD).

The key idea of this model reduction which we will discuss in the following chapter is to use problem-specific ansatz functions $\psi_1, \ldots, \psi_\ell$ instead of the piecewise linear $\varphi_1, \ldots, \varphi_r$ which allow us to choose $\ell \ll r$ with a sufficiently small error for the POD solution to the FE solution. Consequently, this admits

a large increase in computational speed.

Then we will use the a posteriori error estimate we have just derived in order to estimate the difference between the optimal FE solution and the suboptimal POD solution $\|\bar{u}_{ab}^{\ell} - \bar{u}_{ab}\|_U$. We want to emphasize that we can then control the size of the error by the a posteriori error estimate without having to compute the time-consuming high-dimensional solution $\bar{u}_{ab}$.

3 Model reduction using the POD method

This chapter briefly captures the main results on proper orthogonal decomposition (POD) from [5] in order to derive a reduced-order model for the state and adjoint equation in ($\hat{\mathsf{P}}$) and ($\hat{\mathsf{R}}_{ab}$).

3.1 The continuous POD method

For a fixed control $u \in U$ with corresponding state $y = y(u) \in C([0,T];V)$ and adjoint $p = p(u) \in C([0,T];V)$ we define

$$\mathcal{V} = \operatorname{span}\{y(t), p(t) \mid t \in [0,T]\}$$

which is a subspace of V of dimension $d = \dim(\mathcal{V}) < \infty$.

Definition 3.1. *A solution to the following minimization problem (for $\ell < d$)*

$$\begin{aligned} &\min_{\psi_1,\dots,\psi_\ell \in V} \int_0^T \left\| y(t) - \sum_{i=1}^{\ell} \langle y(t), \psi_i \rangle_V \, \psi_i \right\|_V^2 + \left\| p(t) - \sum_{i=1}^{\ell} \langle p(t), \psi_i \rangle_V \, \psi_i \right\|_V^2 \, \mathrm{d}t \\ &\text{s.t. } \langle \psi_i, \psi_j \rangle_V = \delta_{ij} \text{ for } 1 \le i,j \le \ell \end{aligned} \tag{3.1}$$

is called POD basis of rank ℓ.

So determining a POD basis means finding a set of orthonormal functions in V that give the best approximation in the $L^2(0,T;V)$-sense simultaneously to the trajectories y and p. Note that the POD basis depends on the previously chosen control u and – if well chosen – the POD subspace $V^\ell := \operatorname{span}\{\psi_1, \dots, \psi_\ell\}$ captures the main features of the dynamics of the system. Including the adjoint into the POD basis computation is motivated by the convergence results of [6, 7].

We define (for a fixed state and adjoint) the linear and bounded operator $\mathcal{R}\colon V \to \mathcal{V}$ by

$$\mathcal{R}\psi = \int_0^T \langle y(t), \psi \rangle_V \, y(t) + \langle p(t), \psi \rangle_V \, p(t) \, \mathrm{d}t, \quad \psi \in V$$

which is by [5, Lemma 2.12] compact, self-adjoint and non-negative. Moreover, there exists [5, Theorem 2.13] a full orthonormal basis $\psi_1, \ldots, \psi_d \in \mathcal{V}$ and corresponding eigenvalues $\lambda_1 \geq \ldots \geq \lambda_d \geq 0$ satisfying

$$\mathcal{R}\psi_i = \lambda_i \psi_i \quad \text{for } 1 \leq i \leq d$$

which follows from the Riesz-Schauder and Hilbert-Schmidt Theorems [5, Theorem 2.4 and 2.5]. By induction over ℓ one can show [5, Theorem 2.13] that this orthonormal basis coincides with the POD basis and that

$$\int_0^T \left\| y(t) - \sum_{i=1}^{\ell} \langle y(t), \psi_i \rangle_V \, \psi_i \right\|_V^2 + \left\| p(t) - \sum_{i=1}^{\ell} \langle p(t), \psi_i \rangle_V \, \psi_i \right\|_V^2 \, \mathrm{d}t = \sum_{i=\ell+1}^{d} \lambda_i$$

holds for all $\ell \in \{1, \ldots, d\}$.

3.2 The discrete POD method

In the numerical context we are not given whole trajectories of y and p but approximations $y_j^h \approx y(t_j)$ and $p_j^h \approx p(t_j)$ in a finite-dimensional subspace $V^h \subset V$ at certain time instances $0 \leq t_1 < \ldots < t_N \leq T$, e.g. an equidistant time grid.

Suppose for a fixed $N \in \mathbb{N}$ we are given snapshots $y_1^h, \ldots, y_N^h, p_1^h, \ldots, p_N^h \in V^h$. The snapshot subspace

$$\mathcal{V}^N = \operatorname{span}\{\, y_j^h, p_j^h \mid 1 \leq j \leq N\} \subset V^h$$

is a linear subspace of finite dimension $d^N \in \{1, \ldots, 2N\} < \infty$.

Problem (3.1) translates to

$$\begin{aligned} \min_{\psi_1, \ldots, \psi_\ell \in V^h} \; & \sum_{j=1}^{N} \beta_j^N \left(\left\| y_j^h - \sum_{i=1}^{\ell} \left\langle y_j^h, \psi_i \right\rangle_V \psi_i \right\|_V^2 + \left\| p_j^h - \sum_{i=1}^{\ell} \left\langle p_j^h, \psi_i \right\rangle_V \psi_i \right\|_V^2 \right) \\ \text{s.t.} \; & \langle \psi_i, \psi_j \rangle_V = \delta_{ij} \text{ for } 1 \leq i, j \leq \ell \end{aligned} \tag{3.2}$$

with positive weighting parameters β_j^N from the numerical integration, e.g. trapezoidal weights. Hence, for $\ell \in \{1, \ldots, d^N\}$ the task is to find an orthonormal set $\{\psi_i\}_{i=1}^{\ell}$ in V^h that minimizes the mean square error between the snapshots and their ℓ-th partial Fourier sum for all $2N$ snapshots on average. That

means we try to simultaneously approximate all snapshots by an – ideally much smaller – set of orthonormal functions which makes them reasonable ansatz functions for the Galerkin approach.

As for the continous variant the POD basis can be computed by solving an eigenvalue problem. We define the linear operator $\mathcal{R}^N \colon V \to \mathcal{V}^N$ by

$$\mathcal{R}^N \psi = \sum_{j=1}^{N} \beta_j^N \left(\left\langle y_j^h, \psi \right\rangle_V y_j^h + \left\langle p_j^h, \psi \right\rangle_V p_j^h \right) \quad \text{for } \psi \in V$$

which is as well compact, non-negative and self-adjoint [5, Lemma 2.2]. Again follows from the Riesz-Schauder and the Hilbert-Schmidt Theorems that there exist eigenvalues $\lambda_1 \geq \ldots \geq \lambda_{d^N} \geq 0$ and corresponding eigenfunctions $\psi_1, \ldots, \psi_{d^N} \in \mathcal{V}^N \subset V^h$ satisfying

$$\mathcal{R}^N \psi_i = \lambda_i \psi_i \quad \text{for } 1 \leq i \leq d^N \tag{3.3}$$

For each $\ell \in \{1, \ldots, d^N\}$ these eigenfunctions corresponding to the ℓ largest eigenvalues solve (3.2), see [5, Theorem 2.7], thus are a POD basis of rank ℓ, and for the minimal value of the objective in (3.2) holds

$$\sum_{j=1}^{N} \beta_j^N \left(\left\| y_j^h - \sum_{i=1}^{\ell} \left\langle y_j^h, \psi_i \right\rangle_V \psi_i \right\|_V^2 + \left\| p_j^h - \sum_{i=1}^{\ell} \left\langle p_j^h, \psi_i \right\rangle_V \psi_i \right\|_V^2 \right) = \sum_{i=\ell+1}^{d^N} \lambda_i$$

Thinking of how to choose ℓ one might take the ratio of the modelled vs. the total energy contained in the snapshots $y_1^h, \ldots, y_N^h, p_1^h, \ldots, p_N^h$

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{d^N} \lambda_i} \in [0, 1]$$

into consideration. It is not necessary to compute all eigenvalues as their sum by [5, Remark 2.6] equals the sum of the squared norms of all snapshots

$$\sum_{i=1}^{d^N} \lambda_i = \sum_{j=1}^{N} \beta_j^N \left(\left\| y_j^h \right\|_V^2 + \left\| p_j^h \right\|_V^2 \right)$$

As shown in [5, Theorem 2.16] for an equidistant time grid and trapezoidal weights holds

$$\lim_{N \to \infty} \left\| \mathcal{R}^N - \mathcal{R} \right\|_{\mathcal{L}(V)} = 0$$

3.2.1 The discrete POD method in the Euclidean space

Actually, solving the ODE systems for the state and adjoint variable with a time-stepping scheme will give us snapshots of the vector-valued coefficient functions $\mathbf{y}_j^h, \mathbf{p}_j^h \in \mathbb{R}^r$ at the $j = 1, \ldots, N$ time instances in the FE coefficient space $\mathbb{R}^r$, such that

$$y_j^h(x) = \sum_{i=1}^{r} (\mathbf{y}_j^h)_i \, \varphi_i(x) \quad \text{and} \quad p_j^h(x) = \sum_{i=1}^{r} (\mathbf{p}_j^h)_i \, \varphi_i(x) \tag{3.4}$$

are approximations of the state and adjoint at the j-th time instance, respectively. Thus, we will shortly explain how to transform (3.3) to an eigenvalue problem in $\mathbb{R}^r$. Let us collect the snapshots in the so-called snapshot matrix

$$Y = \left[\mathbf{y}_1^h, \ldots, \mathbf{y}_N^h, \mathbf{p}_1^h, \ldots, \mathbf{p}_N^h\right] \in \mathbb{R}^{r \times 2N} \tag{3.5}$$

Since for $1 \leq j \leq \ell$ we can also expand the POD basis functions in the FE basis

$$\psi_j(x) = \sum_{i=1}^{\ell} \Psi_{ij} \, \varphi_i(x) \tag{3.6}$$

with so-called POD basis vectors $\Psi_{:,j}$ gathered in a matrix $\Psi \in \mathbb{R}^{r \times \ell}$. The topology of $V = H^1(\Omega)$ on the FE space V^h corresponds to a topology induced by a weighted inner product on $\mathbb{R}^r$ defined as

$$\langle x, y \rangle_W = x^\top W y \quad \text{for } x, y \in \mathbb{R}^r$$

with the symmetric and positive definite weight matrix

$$W \in \mathbb{R}^{r \times r} \text{ where } W_{ij} = \langle \varphi_i, \varphi_j \rangle_V \tag{3.7}$$

thus $W = M + A$ where M denotes the mass matrix and A the stiffness matrix as in Section 2.4.

So for functions $f, g \in V^h$, i.e. $f = \sum_{i=1}^r \mathbf{f}_i \varphi_i$ and $g = \sum_{i=1}^r \mathbf{g}_i \varphi_i$ with coefficients $\mathbf{f}, \mathbf{g} \in \mathbb{R}^r$ holds

$$\langle f, g \rangle_V = \langle \mathbf{f}, \mathbf{g} \rangle_W$$

which means the POD basis vectors $\Psi_{:,1}, \ldots, \Psi_{:,\ell}$ are W-orthonormal (i.e. Ψ is a W-orthogonal matrix) as the POD basis functions $\psi_1, \ldots, \psi_\ell$ are V-orthonormal. Lastly, define the blockdiagonal (and also diagonal) time-weights matrix

$$\tilde{D} = \begin{pmatrix} D & 0 \\ 0 & D \end{pmatrix} \in \mathbb{R}^{2N \times 2N} \quad \text{for} \quad D = \operatorname{diag}(\beta_1^N, \ldots, \beta_N^N) \in \mathbb{R}^{N \times N} \tag{3.8}$$

consisting of 2 blocks of time-weights from the numerical integration.

Proposition 3.2. *The eigenvalue problem* (3.3) *for the POD basis functions*

$$\mathcal{R}^N \psi_i = \lambda_i \psi_i \quad \textit{for } 1 \leq i \leq \ell$$

is equivalent to the eigenvalue problem

$$Y \tilde{D} Y^\top W \Psi_{:,i} = \lambda_i \Psi_{:,i} \quad \textit{for } 1 \leq i \leq \ell \tag{3.9}$$

for the POD basis vectors with $Y, \Psi, W, \tilde{D}$ *as defined in* (3.5)-(3.8).

Proof. The statement follows from these definitions and matrix computations, see [5, Remark 2.11]. □

Remark 6. There are several possibilities how to compute a POD basis that satisfies (3.9). We will present three ways from [5]. Set the weighted snapshot matrix $\bar{Y} = W^{1/2} Y \tilde{D}^{1/2} \in \mathbb{R}^{r \times 2N}$. Both W and $\tilde{D}$ are symmetric and positive definite, thus their root is well-defined. Then the POD basis Ψ can be computed as follows:

- Solve the $r \times r$ eigenvalue problem

  $$\bar{Y} \bar{Y}^\top u_i = \lambda_i u_i \quad \text{s.t. } u_i^\top u_j = \delta_{ij},\ 1 \leq i, j \leq \ell$$

  for the ℓ largest eigenvalues $\lambda_1 \geq \ldots \geq \lambda_\ell$ and set

  $$\Psi_{:,i} = W^{-1/2} u_i, \quad 1 \leq i \leq \ell$$

  If $r \leq 2N$ this is the smaller eigenvalue problem, but it is necessary to compute $W^{1/2}$ which might be time-consuming depending on the structure of W.

- Solve the associated $2N \times 2N$ eigenvalue problem

  $$\bar{Y}^\top \bar{Y} v_i = \lambda_i v_i \quad \text{s.t. } v_i^\top v_j = \delta_{ij},\ 1 \leq i, j \leq \ell$$

  for the ℓ largest eigenvalues $\lambda_1 \geq \ldots \geq \lambda_\ell$ and set

  $$\Psi_{:,i} = \lambda_i^{-1/2} Y \tilde{D}^{1/2} v_i, \quad 1 \leq i \leq \ell$$

  This avoids the computation of the square root of the weight matrix W because

  $$\bar{Y}^\top \bar{Y} = \tilde{D}^{1/2} Y^\top W Y \tilde{D}^{1/2}$$

  which can be convenient in cases when W is not diagonal whereas the time-weights matrix $\tilde{D}$ is diagonal (and thus the computation of its root fast).

- Compute the singular value decomposition of

$$\bar{Y} = U\Sigma V^\top$$

with orthogonal matrices $U \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{2N\times 2N}$ and a matrix $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{d^N}) \in \mathbb{R}^{r\times 2N}$ containing the singular values for which holds $\sigma_i^2 = \lambda_i$. Again we set

$$\Psi_{:,i} = W^{-1/2} u_i, \quad 1 \le i \le \ell$$

where u_i denotes the i-th column of U. This approach is more stable as taking the product of the weighted snapshot matrices squares the condition number in both eigenvalue problems, but it is also more costly.

In our numerical experiments we will choose the second option, as the problem is small enough and W is non-diagonal as it contains the assembled area integrals. ◊

3.3 Reduced-order modelling for the optimal control problem

Suppose we have computed a POD basis $\{\psi_i\}_{i=1}^{\ell} \subset V^h$ of rank ℓ from snapshots $y^h = \mathcal{S}^h u$ and $p^h = \mathcal{A}^h u$ with N time instances each. Then we define the POD subspace

$$V^{h\ell} = \operatorname{span}\{\psi_1, \ldots, \psi_\ell\} \subset V^h \tag{3.10}$$

3.3.1 The POD Galerkin approximation

We replace the FE Galerkin expansion (2.20) by the POD Galerkin expansion

$$y^{h\ell}(t,x) = \sum_{i=1}^{\ell} \mathbf{y}_i^{h\ell}(t)\psi_i(x) \in V^{h\ell} \quad \text{a.e. in [0,T]} \tag{3.11}$$

and

$$p^{h\ell}(t,x) = \sum_{i=1}^{\ell} \mathbf{p}_i^{h\ell}(t)\psi_i(x) \in V^{h\ell} \quad \text{a.e. in [0,T]} \tag{3.12}$$

with coefficient functions $\mathbf{y}^{h\ell}, \mathbf{p}^{h\ell} \colon [0,T] \to \mathbb{R}^\ell$ in order to find low-dimensional approximations to the state variable y and the adjoint variable p.

Plugging the Galerkin expansion for $y^{h\ell}$ and $p^{h\ell}$ into their weak formulations (2.9) and (2.15) and choosing $V^{h\ell}$ as test function space we obtain the ODE systems

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} M^\ell \dot{\mathbf{y}}^{h\ell}(t) + A^\ell \mathbf{y}^{h\ell}(t) &= B^\ell(u,t) \\ M^\ell \mathbf{y}^{h\ell}(0) &= y_0^{h\ell} \end{aligned} \tag{3.13}$$

and

$$\begin{aligned} -M^\ell \, \dot{\mathbf{p}}^{h\ell}(t) + A^\ell \, \mathbf{p}^{h\ell}(t) &= \alpha_Q \left(y_d^{h\ell}(t) - M^\ell \, \mathbf{y}^{h\ell} \right) \\ \mathbf{p}^{h\ell}(T) &= 0 \end{aligned} \tag{3.14}$$

where

- $M^\ell \in \mathbb{R}^{\ell \times \ell}$ with $M^\ell_{ij} = \langle \psi_i, \psi_j \rangle_H$
- $A^\ell \in \mathbb{R}^{\ell \times \ell}$ with $A^\ell_{ij} = \langle \nabla \psi_i, \nabla \psi_j \rangle_H$
- $B^\ell \colon U \to L^2(0,T;\mathbb{R}^\ell)$ with $B^\ell_i(u,t) = \langle (\mathcal{B}u)(t), \psi_i \rangle_{V' \times V}$
- $y_0^{h\ell} \in \mathbb{R}^\ell$ with $(y_0^{h\ell})_i = \langle y_0, \psi_i \rangle_H$
- $y_d^{h\ell} \in L^2(0,T;\mathbb{R}^\ell)$ with $(y_d^{h\ell})_i(t) = \langle y_d(t), \psi_i \rangle_H$

Using the expansion of the POD basis functions in the FE basis as in (3.6) we can also compute the reduced matrices from the FE matrices as follows:

$$M^\ell = \Psi^\top M \Psi, \qquad A^\ell = \Psi^\top A \Psi, \qquad B^\ell(u,t) = B(u,t)\Psi \tag{3.15}$$

For the initial and the desired temperatures this works analoguously, but we rather again split the solution into

$$y^{h\ell} = \hat{y}^h + \mathcal{S}^{h\ell} u, \quad \text{with } (\mathcal{S}^{h\ell} u)(t) \in V^{h\ell} \text{ a.e. in } [0,T]$$

where the linear operator $\mathcal{S}^{h\ell} \colon U \to W(0,T)$ maps onto the unique weak solution for the homogeneous system (3.13), i.e. $y_0^{h\ell} = 0$. Analoguously for the adjoint

$$p^{h\ell} = \hat{p}^h + \mathcal{A}^{h\ell} u, \quad \text{with } (\mathcal{A}^{h\ell} u)(t) \in V^{h\ell} \text{ a.e. in } [0,T]$$

introducing the operator $\mathcal{A}^{h\ell} \colon U \to W(0,T)$ mapping onto the unique weak solution of (3.14) with $\hat{\mathbf{y}}^{h\ell} = 0$.

Remark 7. We decide to compute $\hat{y}^h$ and $\hat{p}^h$ by solving their high-dimensional models as it has to be performed only once throughout the whole optimization

process of solving $(\hat{P}^h)$. The advantage is that $y^{h\ell}(0) = y_0$ and $p^{h\ell}(T) = 0$, i.e. we don't make an initial approximation error. $\Diamond$

3.3.2 Optimality conditions

Analoguously to Section 2.3 and 2.4 we get

$$\mathcal{B}'\mathcal{A}^{h\ell} = -(\mathcal{S}^{h\ell})'\Theta\mathcal{S}^{h\ell} \in \mathcal{L}(U) \quad \text{and} \quad (\mathcal{S}^{h\ell})'\Xi(y_d - \mathcal{E}\hat{y}^h) = \mathcal{B}'\hat{p}^h$$

which follows by the same arguments as in Lemmas 2.8 and 2.9, since the POD basis for the state and adjoint variable coincide.

Now we can introduce the cost functional for the low-dimensional model as

$$\min \hat{J}^{h\ell}(u) = J(\mathcal{E}(\hat{y}^h + \mathcal{S}^{h\ell}u), u) \quad \text{s.t. } u \in U_{ad}^{int} \tag{$\hat{P}^{h\ell}$}$$

and its relaxation

$$\min \hat{J}^{h\ell}(u) \text{ s.t. } u \in U_{ad}(a,b) \tag{$\hat{R}_{ab}^{h\ell}$}$$

For given $a, b \in \mathbb{Z}^m$ with $a \leq b$ holds that $\bar{u}_{ab}^{\ell}$ is an optimal solution to $(\hat{R}_{ab}^{h\ell})$, iff $(\bar{y}_{ab}^{h\ell}, \bar{p}_{ab}^{h\ell}, \bar{u}_{ab}^{\ell})$ satisfy the first order necessary optimality system

$$\begin{aligned} &\bar{y}_{ab}^{h\ell} = \hat{y}^h + \mathcal{S}^{h\ell}\bar{u}_{ab}^{\ell}, \quad \bar{p}_{ab}^{h\ell} = \hat{p}^h + \mathcal{A}^{h\ell}\bar{u}_{ab}^{\ell}, \quad u_a \leq \bar{u}_{ab}^{\ell} \leq u_b \\ &\left\langle \nabla F(\bar{u}_{ab}^{\ell}) - \mathcal{B}'\bar{p}_{ab}^{h\ell}, u - \bar{u}_{ab}^{\ell} \right\rangle_U \geq 0 \quad \text{for all } u \in U_{ad}(a,b) \end{aligned} \tag{3.16}$$

The proof is analoguous to Theorem 2.10 in Section 2.3.

3.3.3 A-posteriori error estimates

In Chapter 2 we have seen that an optimal control $\bar{u}_{ab} \in U_{ad}(a,b)$ satisfies the variational inequality (2.25) and for any other admissible control holds a perturbed variational inequality with a perturbation function $\xi \in U$, see (2.30).
So after having computed a POD optimal control $\bar{u}_{ab}^{\ell}$ by solving the low-dimensional system (3.16) for a relaxation $(\hat{R}_{ab}^{h\ell})$ we can use this estimate (2.32) with $\tilde{u} = \bar{u}_{ab}^{\ell}$:

$$\left\| \bar{u}_{ab} - \bar{u}_{ab}^{\ell} \right\|_U \leq \Delta_{\text{apost}}(\bar{u}_{ab}^{\ell}) = \bar{\alpha}^{-1} \left\| \xi(\bar{u}_{ab}^{\ell}) \right\|_U$$

to see how well the FE optimal control $\bar{u}_{ab}$ got approximated without having to actually compute it. That means, we can control the error by the POD approximation without solving the costly high-dimensional model. And if the a-posteriori estimator exceeds a certain tolerance, we could think of resolving $(\hat{R}_{ab}^{h\ell})$ with more POD basis functions, i.e. updating the POD basis. Remember that the POD basis depends on the initially chosen control, so perhaps if that was a bad choice one could compute a new POD basis from a more suitable control (which is costly as it requires full state and adjoint computations).

In the next chapter we will discuss on how to treat the integer control variables and reestablish the integrality constraints which we so far ignored.

4 Mixed-integer nonlinear programming

This chapter reestablishes the integrality constraint for the time-independent boundary control and therefore discusses the branch-and-bound algorithm. Generally speaking, solving a mixed-integer programming is to find

$$z_{MIP} = \min f(u) \text{ s.t. } u \in U_{ad}^{int} \tag{MIP}$$

where the control u is supposed to be in the mixed-integer set U_{ad}^{int}.

Remark 8. In our setting we will deal with $f = \hat{J}^h$ or $f = \hat{J}^{h\ell}$ whose underlying system dynamics are in fact both linear because of the approximation of y in the boundary condition (see Remark 2) which makes them much faster to solve. Nevertheless, we will stick to the more general context of nonlinear programming and point out its difficulties where they would arise. $\Diamond$

Recall that $u = (u^C, u^I) \in U = L^2(0,T;\mathbb{R}^n) \times \mathbb{R}^m$ with time-dependent distributed control u^C and the boundary control u^I which is supposed to be integer-valued. The mixed-integer set U_{ad}^{int} is as previously in Chapter 2 defined by

$$U_{ad}^{int} = \{u \in U \,|\, u_a \leq u \leq u_b, u^I \in \mathbb{Z}^m\} \tag{4.1}$$

for bilateral constraints $u_a, u_b \in U$. As mentioned before in Remark 4 the notation of the relaxed set of admissible controls includes the lower and upper constraints $a, b \in \mathbb{Z}^m$ on the integer variable

$$U_{ad}(a,b) = \{u \in U \,|\, a \leq u^I \leq b\}$$

as these will change during the optimization process in the branch-and-bound algorithm.

Definition 4.1. *For $a, b \in \mathbb{Z}^m$ the relaxed problem or relaxation is given by omitting the integrality constraints*

$$z^{R(a,b)} = \min f(u) \ \textit{s.t.}\ u \in U_{ad}(a,b) \tag{$R_{ab}^\star$}$$

From Chapter 2 and 3 we know that for both our choices of f a relaxed problem (i.e. $(\hat{R}^h_{ab})$ or $(\hat{R}^{h\ell}_{ab})$) either has a minimizer $u^\star_{ab} \in U_{ad}(a,b)$ for all $a, b \in \mathbb{Z}^m$ if $a \leq b$ or it is infeasible, i.e. no solution exists, as it can happen in the branch-and-bound algorithm that $a_i > b_i$ for an index $i \in I$ and thus the feasible set of integer controls is empty.

Clearly, there is no evidence that rounding certain components of the minimizer to integer values will yield a good solution of (MIP). Hence, in the following chapter we will discuss the branch-and-bound method which is a very common approach for mixed-integer problems.

4.1 The branch-and-bound algorithm

The branch-and-bound algorithm starts by solving the relaxation of the original mixed-integer nonlinear programming, i.e. $(R^\star_{ab})$ with $a = u^I_a$ and $b = u^I_b$.

If this relaxation is infeasible, the (MIP) itself is infeasible. If we find a solution $u^\star_{ab}$ which already satisfies the integrality constraints, this solution solves (MIP) and we can stop. Otherwise, the solution $u^\star_{ab}$ violates the integrality condition, i.e. there exists an index $i \in I$ for which $(u^{\star I}_{ab})_i \notin \mathbb{Z}$. Then we branch on this variable u^I_i by dividing the feasible region into two subproblems: $(R^\star_{a\tilde{b}})$ and $(R^\star_{\tilde{a}b})$ with

$$\tilde{a}_i := \lceil (u^{\star I}_{ab})_i \rceil \geq (u^{\star I}_{ab})_i, \qquad \tilde{a} = (a_1, \ldots, a_{i-1}, \tilde{a}_i, a_{i+1}, \ldots, a_m)$$

$$\tilde{b}_i := \lfloor (u^{\star I}_{ab})_i \rfloor \leq (u^{\star I}_{ab})_i, \qquad \tilde{b} = (b_1, \ldots, b_{i-1}, \tilde{b}_i, b_{i+1}, \ldots, b_m)$$

so the current solution $u^\star_{ab}$ will no longer be feasible in neither of the resulting two subproblems. Then we continue by solving one of the open subproblems and repeating these steps: if its solution is integer-valued continue with solving the next open subproblem, if it still violates the integrality constraints branch as before and add two resulting subproblems to the heap of open problems. So this algorithm gives a binary tree structure whose nodes are defined by a set of bounds which determine a subproblem.

Recall the definition of a (feasible) relaxation $(R^\star_{ab})$ where the objective of the solution $u^\star_{ab}$ is denoted as $z^{R(a,b)}$, so it is the minimal value of f in this particular subregion $U_{ad}(a,b)$. If $u^{\star I}_{ab} \notin \mathbb{Z}^m$ then $z^{R(a,b)}$ is a lower bound for the minimum value in this feasible subregion and thus all further subproblems arising from this subproblem, i.e. the branch arising from this node. Otherwise, if $u^{\star I}_{ab} \in \mathbb{Z}^m$ satisfies all integrality constraints then $z^{R(a,b)}$ is a global upper bound for the

minimum value z_{MIP} of the mixed-integer problem (MIP) as we might find a better solution in one of the subproblems which are still to solve.

The advantage of this approach is that subproblems whose lower bound $z_{low}^{R(a,b)}$ is already greater than the global upper bound z_{up} and also infeasible subproblems can be neglected, speaking in terms of the tree structure: this node (and the branch which would have arosen from it) can be pruned. On the other hand the branch-and-bound algorithm can be very consuming if many variables are required to be integers or if the nonlinear problems that have to be solved are complicated.

Algorithm 1 [branch-and-bound method]

1: Set the global upper bound $z_{up} = \infty$ and initialize the heap of open problems $\mathcal{H} = \emptyset$;
2: Add the root node relaxation $(R_{ab}^{\star})$ with $a = u_a^I$, $b = u_b^I$ to $\mathcal{H}$;
3: **while** $\mathcal{H} \neq \emptyset$ **do**
4: Select a problem $(R_{ab}^{\star})$ from $\mathcal{H}$ and set $\mathcal{H} := \mathcal{H} \backslash (R_{ab}^{\star})$;
5: Solve relaxation $(R_{ab}^{\star})$ and let the solution be $u_{ab}^{\star} = (u_{ab}^{\star C}, u_{ab}^{\star I})$;
6: **if** $(R_{ab}^{\star})$ is infeasible **then**
7: Node can be pruned;
8: **else if** $f(u_{ab}^{\star}) > z_{up}$ **then**
9: Node can be pruned;
10: **else if** $u_{ab}^{\star I}$ is integral **then**
11: Update incumbent solution: $z_{up} = f(u_{ab}^{\star})$, $u^{\star} = u_{ab}^{\star}$;
12: **else**
13: Branch on a variable $i \in I$ with $(u_{ab}^{\star I})_i \notin \mathbb{Z}$ and add the two resulting subproblems to $\mathcal{H}$.
14: **end if**
15: **end while**
16: **return** optimal solution $u^{\star}$ or error message (if no feasible solution is found).

The algorithm terminates after a finite number of steps either with an optimal solution or with an indication that the problem is infeasible, because the set of admissible boundary controls is finite and a relaxation always either has a unique optimal solution or is infeasible.

In the following section we will give details for line 13 in Algorithm 1 how to choose a variable to branch upon (in case there is more than one violating integrality) and in the subsequent section we will present strategies for line 4 in Algorithm 1 how to decide for the next problem from the heap of open problems.

4.2 Selection of the branching variable

Suppose we solved a particular node in the branch-and-bound tree whose solution $u^\star_{ab}$ does not satisfy all integrality constraints and we want to choose the next variable to branch on. As it is not a-priori clear which choice is the best one there exist several approaches. Let $I_c \subset I$ denote the set of all candidate branching variables, for example the set of all indices of violated integrality constraints $I_c = \{i \in I \,|\, (u^{\star I}_{ab})_i \notin \mathbb{Z}\}$.

As a first idea one could choose the variable from the candidate set which violates the integrality condition the most, i.e.

$$\underset{i \in I_c}{\operatorname{argmax}} \{\min((u^{\star I}_{ab})_i - \lfloor (u^{\star I}_{ab})_i \rfloor, \lceil (u^{\star I}_{ab})_i \rceil - (u^{\star I}_{ab})_i)\}$$

which is called **maximum fractional branching**. Unfortunately, according to [1] this is not efficient in practical computing as it performs about as well as randomly choosing a variable from the candidate set.

A second and more successful option is to estimate the change in the lower bound after branching, i.e. the difference between child and parent objective. We are interested in increasing it as much as possible because we can prune a node whenever its lower bound is greater than the current global upper bound. Thus, for every candidate $i \in I_c$ we define D_i^- and D_i^+ which estimate the increase in the lower bound after branching down and up on the i-th variable, respectively. Now by

$$s_i := \mu \min(D_i^-, D_i^+) + (1 - \mu) \max(D_i^-, D_i^+) \tag{4.2}$$

we combine the two estimates where $\mu \in [0, 1]$ is a fixed parameter typically close to 1 (see [1]) and choose the variable that maximizes the score function

$$\underset{i \in I_c}{\operatorname{argmax}} \, s_i$$

We will discuss two strategies of computing such estimators D_i^- and D_i^+ and an attractive hybrid of these two.

Strong branching.
For all candidates we solve both child nodes, that means we need to solve $2 \cdot |I_c|$ nonlinear problems (NLPs) which can obviously be expensive depending on the nonlinearity of the problems and the amount of candidates. Note that if one subproblem is infeasible we can tighten the bound and resolve the parent

node. Otherwise let z_i^- and z_i^+ be the objectives of the solutions resulting by branching down and up on the i-th variable. Then we can compute the estimators as the actual change

$$D_i^- = z_i^- - z \quad \text{and} \quad D_i^+ = z_i^+ - z$$

where $z \in \mathbb{R}$ denotes the parent objective.

This approach can yield a significantly smaller number of nodes in the branch-and-bound tree, but it might be slow if there are a lot of branching candidates and thus many nonlinear relaxations have to be solved.

Pseudocosts branching.
Unlike the strong branching this approach does not compute the increase in the lower bound, but tries to estimate it based on pseudocosts from previous branching decisions.

For every integer variable let n_i^+, n_i^- denote the number of times we have already solved a node created from branching up or down on the i-th variable, respectively. Whenever solving a node we update the corresponding pseudocosts, for an up-child

$$p_i^+ = \frac{z_i^+ - z}{\lceil \hat{u}_i^I \rceil - \hat{u}_i^I} + p_i^+, \quad n_i^+ = n_i^+ + 1$$

or for a down-child

$$p_i^- = \frac{z_i^- - z}{\hat{u}_i^I - \lfloor \hat{u}_i^I \rfloor} + p_i^-, \quad n_i^- = n_i^- + 1$$

where $z \in \mathbb{R}$ denotes the objective of the parent node at its optimal solution $\hat{u} = (\hat{u}^C, \hat{u}^I) \in U$, and $z_i^+, z_i^- \in \mathbb{R}$ denote the values of the objective at the minimizers of the child nodes each. So the change in the lower bound is divided by the fractional part of the variable that was branched upon to create this node.

We use the pseudocosts for updating the per unit change in the objective when we would branch on the i-th variable, so whenever we need to make a branching decision we compute

$$D_i^+ = (\lceil \hat{u}_i^I \rceil - \hat{u}_i^I)\,\frac{p_i^+}{n_i^+} \qquad \text{and} \qquad D_i^- = (\hat{u}_i^I - \lfloor \hat{u}_i^I \rfloor)\,\frac{p_i^-}{n_i^-}$$

from which we can then compute the score s_i (4.2) for all branching candidates and choose the index which maximizes the score.

The initialization of the pseudocosts is typically done by strong branching.

Reliability branching.
This is a popular combination of the previous two branching strategies. It starts with the strong branching method while computing the pseudocosts in each step until a certain number of decisions was performed, i.e. n_i^+ or n_i^- are greater than a treshold τ, for example $\tau = 5$ could be a typical choice [1]. The assumption is that enough information about the changes in the lower bounds is gathered in the pseudocosts by then, such that upcoming decisions based on them seem reliable.

4.3 Selection of the branching node

After selecting a variable to branch upon and adding the two resulting child problems to the heap of open problems, we need to figure out which node to treat next. We present two strategies, depth-first search and best-bound search and a combination of the two.

Depth-first search selects the deepest node in the tree, in our case this will be the node that was added last to the heap of open problems. This strategy keeps the amount of open problems as small as possible, but unfortunately it can perform poorly if no upper bound z_{up} is found before the algorithm terminates or if it is found very late. Thus, we might explore many nodes which could have been pruned if an integer solution (whose objective gives a global upper bound z_{up}) was found earlier during the algorithm if their lower bound was actually larger than z_{up}.

Best-bound search selects the node with the best (i.e. lowest) lower bound in the hope to find an integer solution with an also small objective which then

gives a small z_{up}. As it is a lower bound to the objective of an integer solution in this feasible subregion, choosing the best lower bound gives a lower bound to the objective of all integer solutions (that can still be found) and thus to z_{MIP}. Consequently, the node with the best lower bound will always be explored during the algorithm because its lower bound is smaller than any z_{up} which could still be found. So, the node could not have been pruned if we had solved another node first anyways. As a result, this minimizes the number of explored nodes for a fixed number of branching decisions. On the other hand there will be more open problems that need to be stored and usually it does not find an integer solution before the end of the search because this z_{up} then usually dominates all other open problems, i.e. their lower bound is larger and they can be pruned. Hence, if the solution time is limited, we might not find any feasible point at all.

Two-phase method starts with the depth-first strategy until one or a small number of integer solution(s) and thus feasible point(s) for (MIP) were found. Then it continues by the best-search strategy. If the search tree in this second phase becomes too large one could also switch back to the depth-first strategy to keep the number of open problems manageable.

5 Numerical experiments

Post approval additional remark. Meanwhile, we figured out that the relaxed problems were not always optimally solved with the projected Armijo-backtracking strategy. Replaced with the standard Armjio condition based on the gradient, the a-posteriori estimator performed much better and probably, the later observed unexpected negative values in the estimators D^+ and D^- will then not occur anymore.

In this chapter we will test the branch-and-bound algorithm for two data samples for the outside temperature y_a in the Neumann boundary condition of the heat equation (2.4) each with one or four individual underfloor heating tiles.

Throughout the whole chapter we will consider a horizontal cross-section of a squared room $\Omega = [0,1]^2$ with two interior walls and two outer walls and a window in the middle of one of the latter as illustrated in the following figure.

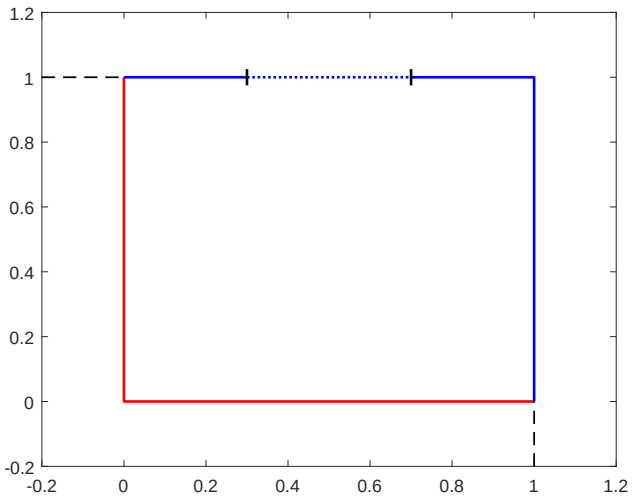


Figure 1: Illustration of the domain Ω with interior walls (red) and outer walls (blue) and a window (dotted blue, $0.3 \leq x_1 \leq 0.7, x_2 = 1$). The black dashed lines indicate further rooms in the house.

The red boundary segments are interior walls, i.e. form the set Γ_0, where we assume negligible heat transition as the other rooms should be of similar warmth as the one we study. The blue boundary segments are outer walls in counterclockwise order, i.e. Γ_1 refers to the long boundary segment around the up-right corner, Γ_2 corresponds to the window and Γ_3 denotes the short outer wall. For these we want to find optimal choices of insulation material: $u^I \in \mathbb{Z}^3$. The material shall be more expensive for the window and the heat transition will be increased there. This is represented by

$$\alpha_1^I = 0.1, \quad \alpha_2^I = 0.2, \quad \alpha_3^I = 0.1 \text{ and } \chi_j^I(s) = -\alpha_j^I \text{ for } s \in \Gamma_j$$

Recall the mixed-integer optimal control problem (2.1) - (2.4).

For the boundary control we require

$$u_a^I = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \leq u^I \leq \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} = u_b^I = \omega$$

where the low values correspond to a very good and thus expensive insulation (modelled in the cost functional by subtracting the reference control ω, see Remark 2 in Chapter 2). Note that a window cannot be insulated as well as a wall. For the heating tiles we allow values between 0 (not heating) and 10 (heating at maximum) varying over time, so

$$0 \leq u_i^C(t) \leq 10 \quad \text{a.e. in } [0,T] \quad \text{for } 1 \leq i \leq n$$

Furthermore, we set $\alpha_Q = 20$ (unless specified otherwise in Remark 9 and 10). For the desired inside temperature we will fix $y_d = 22°\text{C}$ everywhere at all times and the approximation to the inside temperature is given by $y_b = 22°\text{C}$. The initial inside temperature shall be $y_0 = 20°\text{C}$.

In the partial differential equation we choose the thermal diffusity coefficient $c = 0.5$.

5.1 Implementational aspects

The implementation was done in MATLAB Release 2016a and all test runs were performed on a 2.5 GHz Intel inside core i5 processor with 8 GB RAM.

- **Spatial discretization**
  The domain Ω is discretized by the MATLAB routine `initmesh` with piecewise linear finite elements with a maximal mesh size $h_{\max} = 0.05$ which yields $r = 721$ vertices.

- **ODE solver**
  We consider $T = 1$ day ($= 24$ hours) with $N = 100$ equidistant time steps. The time stepping will either be performed by Rannacher Smoothing (i.e. four half-sized implicit Euler steps followed by regular Crank-Nicholson steps) or by the implicit Euler method.

- **OCP solver**
  All arising optimal control problems are solved by the projected Newton-CG method with maximal 10 iterations in the optimization loop, maximal 20 iterations in the conjugate gradient loop (for finding the next search direction) and maximal 10 iterations in the Armijo backtracking strategy (for finding the step length). The code is taken from Stefan Trenz who wrote it in the course of his PhD.

- **POD basis computation**
  The relaxation of the mixed-integer optimal control problem (i.e. the root node in the branch-and-bound tree) is solved without model reduction. The corresponding optimal state and adjoint form the snapshot matrix from which the POD basis is computed as described in variant 2 in Remark 6 with the MATLAB routine `eig` for the full eigenvalue decomposition, even though we only need the ℓ vectors corresponding to the ℓ largest eigenvalues. This is more efficient than e.g. `eigs` because the product of the weighted snapshot matrix is dense and comparably small $2N \times 2N = 200 \times 200$ whereas `eigs` is more suitable to compute a few eigenvalues of large sparse matrices (see [9] in the MATLAB documentation).

  We measure the error in orthogonality of the rank ℓ POD basis by

  $$e^{\Psi} = \left\| \Psi^{\top} W \Psi - I \right\|_{\mathbb{R}^{\ell}}$$

  in order to monitor the stability of the computation. If the error gets too large one should consider the more costly, but more stable way of computing a POD basis from the singular value decomposition as described in variant 3 in Remark 6 in Section 3.2.1.

- **Accuracy**
  For comparison we compute the full high-order FE solution $\bar{u}_{ab}$ for each node, so we can compute

$$e^u_{abs} = \left\|\bar{u}_{ab} - \bar{u}^\ell_{ab}\right\|_U \qquad \text{and} \qquad e^u_{rel} = \frac{\left\|\bar{u}_{ab} - \bar{u}^\ell_{ab}\right\|_U}{\|\bar{u}_{ab}\|_U}$$

for each of the solved nodes where the integral in the U-norm is numerically approximated (again) by trapezoidal time-weights.

- **Node numbering**
  Since MATLAB itself does not provide tree structures we will label each node with an index where to store it in the heap of open problems which is realized as a standard array in our implementation. The root node will be stored at index 1. If we store a node at an index $k \in \mathbb{N}$ then its down-child will be stored at $2k$, the up-child at $2k+1$ and its parent can be found at the index $\lfloor \frac{k}{2} \rfloor$. Moreover, in the plots which we will study later we will sort the nodes in the order they were treated, so we can define a bijective map

  $$q\colon \mathbb{N} \to \mathbb{N},\ k \mapsto q(k)$$

  which maps an index k in the tree to its corresponding node number in the queue of explored nodes.

5.2 Test 1: May 3rd, n = 1

The outside temperature y_a is created from hourly measurements during one day (24 hours, from midnight till midnight) which are interpolated linearly on the time grid. The measurements are taken from the online archive of measurements and climate data [8]. As first example we take May 3rd, 2017 in Constance, Germany, as depicted in Figure 2. The average day temperature is 11.2°C.

We consider $n = 1$ underfloor heating tile acting on the whole domain Ω and set $\alpha^C = 0.3$. The time-stepping is performed by Rannacher Smoothing. The branch-and-bound algorithm will be performed using the strong branching method for selecting branching variables and the two-phase method for node selection. Without model reduction it requires 78.15 seconds to compute an optimal solution $\bar{u}$, where 7 nodes and 15 linear-quadratic optimal control problems (LPs) were solved. The optimal integer-valued boundary control

$$\bar{u}^I = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$$

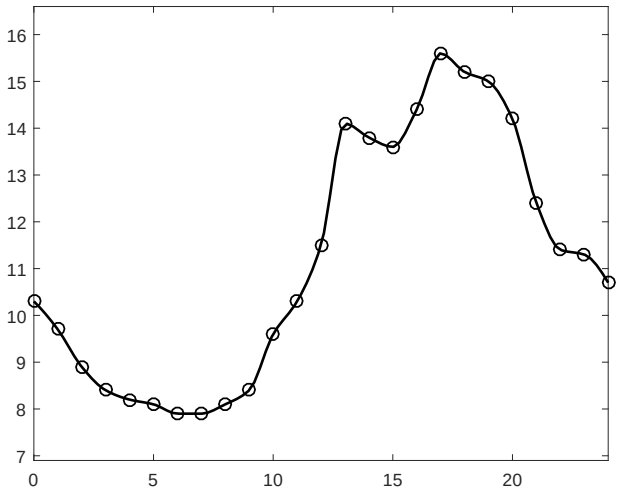


Figure 2: Test 1 and 2: Temperature measurements (circles) and linear interpolation y_a from midnight (0h) till midnight (24h) on May 3rd, 2017 in Constance, Germany

is a quite good insulation: on the long boundary segment Γ_1 and for the window Γ_2 we choose the best available options whereas for the short boundary segment Γ_3 we can take a cheaper one. The choice of insulation materials for the walls might also be influenced by the length of the boundary segments that is not considered in the cost functional. Γ_1 is much longer than Γ_3 and thus needs a better insulation as otherwise much more heat would get lost than when choosing a cheap insulation for Γ_3.
The distributed control $\bar{u}^C$ is displayed in Figure 3. In the beginning of the day we heat at maximum to heat up from the colder initial temperature 20°C. As we get close to the desired temperature of 22°C the heating rapidly decreases. From about 10 o'clock on the outside temperature rises and we have to heat less to maintain the desired temperature despite the heat transition.

Because of the high computational effort necessary to solve this comparably small problem ($n = 1, m = 3$) we test the reduced-order model approach with the POD method which gives the result within 20.72 seconds where also 7 nodes and 15 LPs were solved. Note that a large part of the computational time is still consumed by the full solving (without model-reduction) of the root node in order to compute the POD basis from the resulting optimal control for the relaxation of the original mixed-integer optimal control problem. The error between the high-dimensional and POD solution is very small $e^u_{abs} = 1.6 \cdot 10^{-19}$,

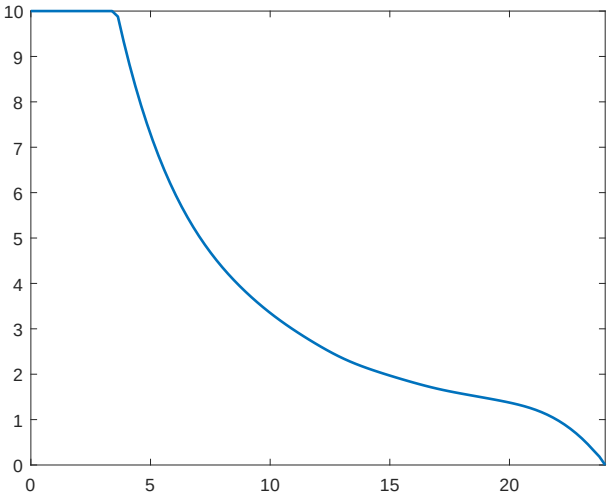


Figure 3: Test 1: Distributed time-dependent part of the optimal control $\bar{u}^C$ during the day (in hours).

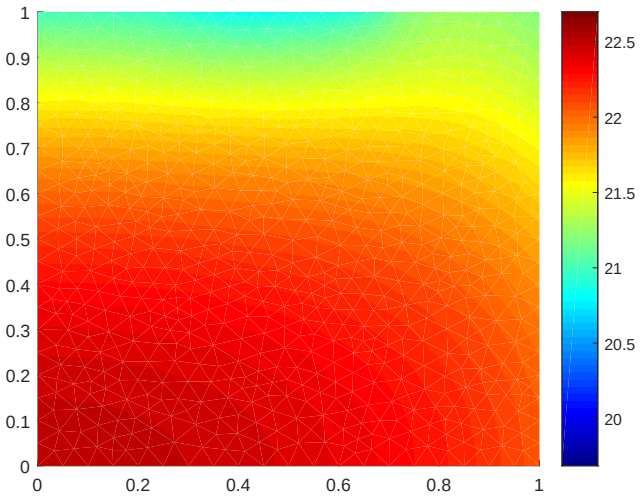


Figure 4: Test 1: Temperature $y(\cdot, T)$ for the optimal control $\bar{u}$ at final time T.

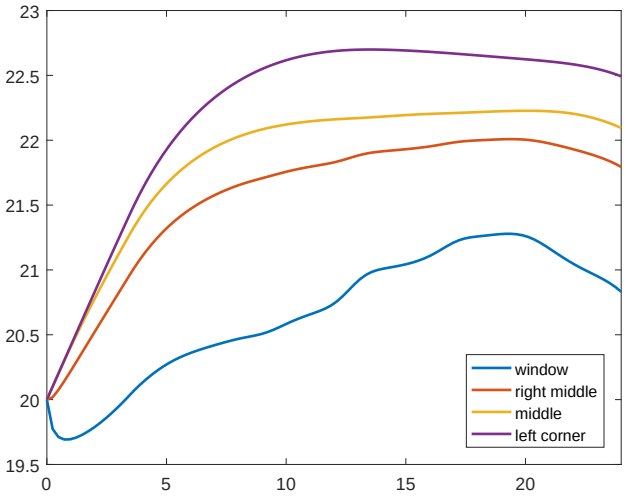


Figure 5: Test 1: Temperature $y(x, \cdot)$ for optimal control $\bar{u}$ at certain points in the room: near the window ($x = (0.5222, 1)$, blue), in the middle of the right outer wall ($x = (1, 0.5)$, red), approximately in the middle of the room ($x = (0.5082, 0.4964)$, yellow) and in the lower-left corner ($x = (0, 0)$, purple).

$\bar{u}^{\ell I} = \bar{u}^I$ and $\bar{u}^{\ell C}$ looks like $\bar{u}^C$ in Figure 3. Also the corresponding state behaves like in Figure 4 and Figure 5.

We compare the suboptimal POD solutions $\bar{u}^\ell_{ab}$ to the full solution computed with the finite element method $\bar{u}_{ab}$ for each node in the branch-and-bound-tree and plot the actual error alongside the a-posteriori estimator as discussed in Section 3.3.3 in Figure 6. The a-posteriori estimator bounds, but usually overshoots the actual error by several orders of magnitude. Perhaps one could investigate this in the near future and try to find sharper error estimations. We decided to not show the actual relative error since it is even smaller.

The idea is to resolve a node when its a-posteriori estimate gets too large, e.g. with an enlarged POD basis by two or perhaps five more basis vectors. Unfortunately here, this barely changes neither the estimator nor the solutions $\bar{u}^\ell_{ab}$, as we can see in Table 1.

As a tolerance we chose to resolve when the a-posteriori estimator got larger than 1, meaning we resolved nodes 3-7 each with the enlarged rank $\ell = 7$ POD basis. The second column of Table 1 shows the a-posteriori estimator for $\bar{u}^\ell_{ab}$ for $\ell = 5$, the third column for the resolved $\bar{u}^\ell_{ab}$ with $\ell = 7$. We also need to solve 7 nodes, but 25 LPs (as many of the branching candidates' children got

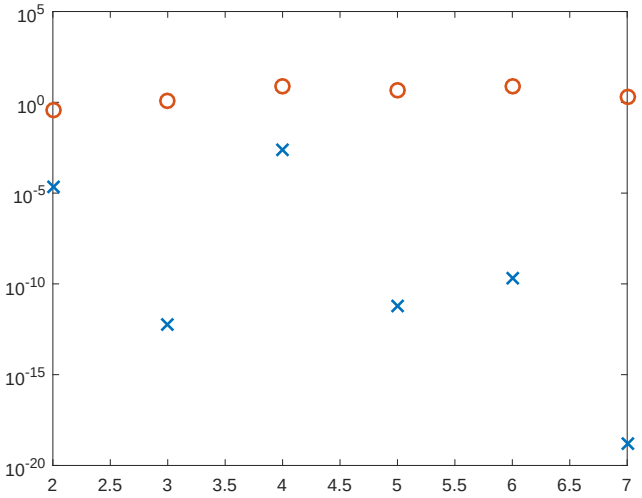


Figure 6: Test 1: Error between high- and low-dimensional optimal solution $\|\bar{u} - \bar{u}^\ell\|_U$ (blue crosses) and a-posteriori estimator (red circles) at each node in the order they were treated.

resolved, too). Choosing an additional 5 instead of 2 basis vectors (so $\ell = 10$) gives the same results as in the third column.

node	$\Delta_{\text{apost}},\ \ell = 5$	$\Delta_{\text{apost}},\ \ell = 7$
2	0.3945	-
3	1.2291	1.2285
4	7.8301	7.8263
5	4.4690	4.4668
6	7.7493	7.7455
7	1.8531	1.8522

Table 1: Test 1: The second column shows the a-posteriori error estimates for the nodes in the order they were solved with $\ell = 5$ and if that is larger than the tolerance 1 we resolved the nodes for $\ell = 7$ and display the updated a-posteriori error estimate in the third column.

Consequently, we decided to not resolve any nodes in the following examples and leave this question open for the future.

Nevertheless, after having seen that $\|\bar{u}_{ab} - \bar{u}_{ab}^\ell\|_U$ is pretty small, we look at $|\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}_{ab}^\ell)|$ in Table 2 which is in most cases also sufficiently small and shows the same peaks as the error of the controls does.

node	$\lvert\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}^\ell_{ab})\rvert$
2	0.0086
3	7.6982e-07
4	0.0140
5	2.3872e-06
6	5.7794e-05
7	4.4076e-10

Table 2: Test 1: Difference in the cost function values for the optimal and POD optimal control, i.e. $|\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}^\ell_{ab})|$, at each node 2-7 on the order they were treated according to the two-phase node selection method.

Looking at the POD basis elements we may recognize the main characteristics of the system. The first one, Figure 7, shows the warm lower-left corner where the inner walls meet and the colder upper half of the room, especially near the window. When multiplying the POD basis vector in Figure 8 by -1 – this does not affect the orthogonality of Ψ – we again see the cold areas in the room near the window. The third POD basis element, Figure 9, shows warmth near the up-right corner where the best available insulation was chosen. Those three elements reflect the structure of the room. The further ones already correspond to fairly small eigenvalues especially compared to the largest one. Actually, we have:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_5 \end{pmatrix} = \begin{pmatrix} 440.2402 \\ 4.5037 \\ 0.9078 \\ 0.0176 \\ 0.0007 \end{pmatrix}$$

This is quite typical for linear-quadratic problems where a few basis vectors can already represent the main characteristics of the problem.

5.2.1 A close look on the branch-and-bound algorithm

The following carefully examines each step in the branch-and-bound algorithm explained in Chapter 4 to gain a profound understanding of the algorithm. It starts with the root node (with index $k = 1$, for the numbering of the nodes in the tree, see again Section 5.1), i.e. the relaxation of the original mixed-integer problem, which is solved fully without model-reduction to the optimal solution $\bar{u}_{ab} \in U$ with

$$\bar{u}^I_{ab} = \begin{pmatrix} 1.0000 \\ 2.3053 \\ 4.0333 \end{pmatrix} \qquad \text{(node 1, k=1)}$$

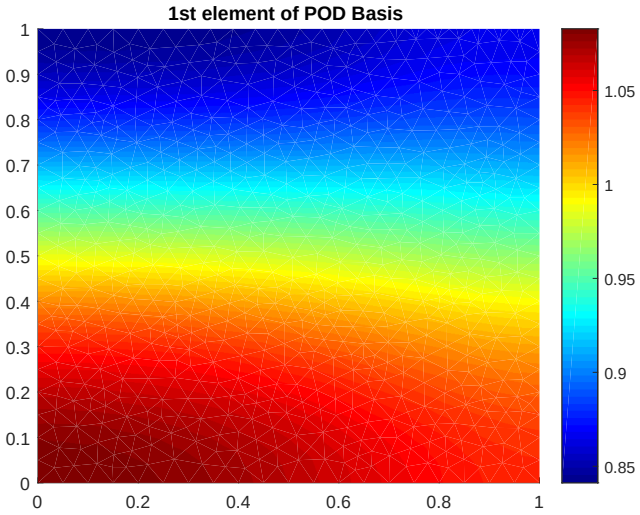


Figure 7: First POD basis element on the domain Ω.

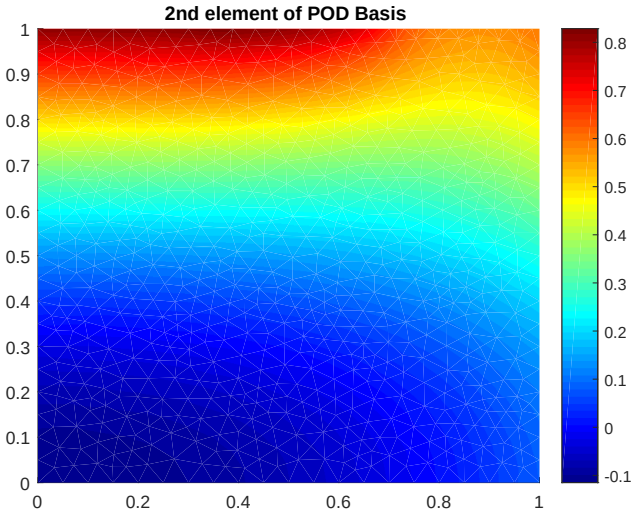


Figure 8: Second POD basis element on the domain Ω.

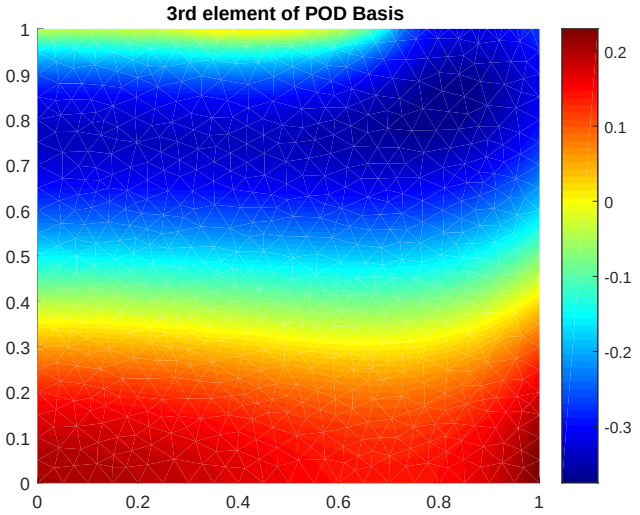


Figure 9: Third POD basis element on the domain Ω.

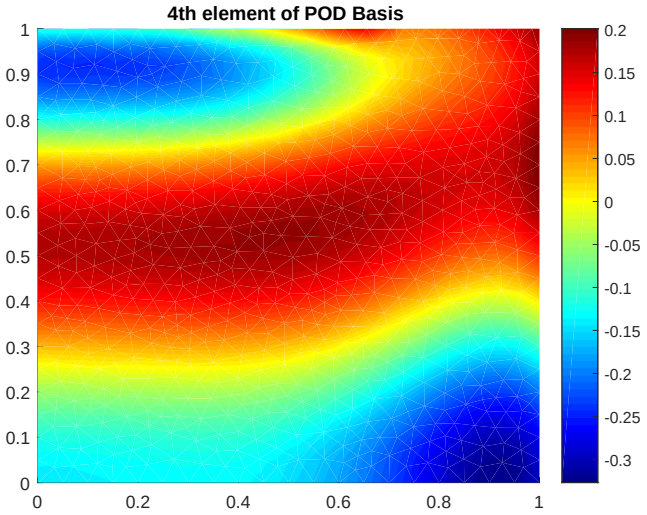


Figure 10: Fourth POD basis element on the domain Ω.

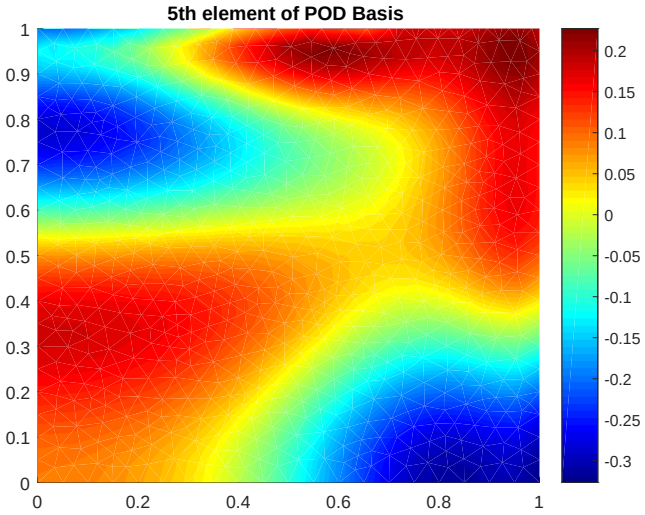


Figure 11: Fifth POD basis element on the domain Ω.

From the corresponding state $\bar{y}_{ab}^h$ and adjoint $\bar{p}_{ab}^h$ we compute a rank $\ell = 5$ POD basis Ψ via solving the smaller $2N \times 2N$ eigenvalue problem (variant 2 in Remark 6). Its error in orthogonality is $e^\Psi = 2.8973 \cdot 10^{-12}$ and the ratio of the energy contained in these five POD basis vectors to the total energy is $\mathcal{E}(5) = 1 - 2.1302 \cdot 10^{-7}$.

The second and third component of $\bar{u}_{ab}^I$ violate the integrality constraints, so they form the set of branching candidates $I_c = \{2, 3\}$ and the cost functional value gives a lower bound $z_{\text{low}}^{(1)} = 22.1422$ as described in Section 4.1. It is valid in the branch arising from this node, so here in all nodes because the current node is the root node.
The variable to branch upon is chosen from the set of candidates according to the strong branching method, i.e. we solve all child problems and compute their changes in the lower bounds D^+ and D^- from which we compute the score s with the parameter $\mu = 0.9$ (a typical value according to [1]) as explained in Section 4.2:

$$D^+ = \begin{pmatrix} 0.1910 \\ 0.0389 \end{pmatrix} \quad D^- = \begin{pmatrix} -0.1092 \\ -0.0080 \end{pmatrix} \quad s = \begin{pmatrix} -0.0792 \\ -0.0034 \end{pmatrix}$$

Technically, there should be no negative values in these estimators because all feasible points in a child node were as well feasible in the parent node, thus a solution with a lower objective in a child node should have been found already during solving its parent node. This might occur due to the fact that we are not in exact arithmetics, but stop when satisfying certain tolerances.

And maybe it is also influenced by our problem not having a clearly unique solution leading to a lot of small steps in the optimization.

Nevertheless, here we branch upon variable u_3^I and create subproblem $k = 2$ requiring $u_3^I \leq 4$ and subproblem $k = 3$ requiring $u_3^I \geq 5$. We also refer to them as the down- and up-child as they arise from branching down and up, respectively. Since no integer solution has been found yet, the two-phase method is still in its first phase, the depth-first-search, which selects the last added node as the next. In our case this is the up-child with the index $k = 3$ in the tree: the relaxation $(\hat{R}_{ab}^{h\ell})$ with

$$a = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix} \quad (\bar{u}^\ell)^I = \begin{pmatrix} 1.0000 \\ 2.0397 \\ 5.0000 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} \qquad \text{(node 2, k=3)}$$

whose POD optimal solution was already computed during the previous strong branching decision. The second component of $(\bar{u}^\ell)^I$ still violates the integrality condition, so we branch up and down on the second variable u_2^I which adds the open problems with indices $k = 6$ and $k = 7$ to the heap. Again the up-child $(\hat{R}_{ab}^{h\ell})$ is chosen by the two-phase method as next node:

$$a = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} \quad (\bar{u}^\ell)^I = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} \qquad \text{(node 3, k=7)}$$

This node gives an integer solution and hence its objective a global upper bound $z_{\text{up}} = 22.6947$ for the objective z_{MIP} of the mixed-integer problem. Now we get to the second phase of the two-phase method: we switch to the best-bound strategy which selects the node with the best lower bound. Consequently, this will be the node at index $k = 2$, because its $z_{\text{low}}^{(2)} = 22.1342 < 22.1710 = z_{\text{low}}^{(6)}$ of the node stored at index $k = 6$ which is currently the only other open problem in the heap.

$$a = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad (\bar{u}^\ell)^I = \begin{pmatrix} 1.0000 \\ 2.3052 \\ 3.9999 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 10 \\ 4 \end{pmatrix} \qquad \text{(node 4, k=2)}$$

Its optimal solution as well violates $(\bar{u}^\ell)_2^I, (\bar{u}^\ell)_3^I \in \mathbb{Z}$, so we solve all child problems that would arise from branching upon these candidates and compute the estimators and the score.

$$D^+ = \begin{pmatrix} 0.1990 \\ -0.0000 \end{pmatrix} \quad D^- = \begin{pmatrix} -0.1027 \\ -0.0621 \end{pmatrix} \quad s = \begin{pmatrix} -0.0725 \\ -0.0559 \end{pmatrix}$$

The second candidate, i.e. third variable maximizes the score and will be branched upon (adding nodes 4 and 5). From the three open problems on the heap: at index $k = 4$ with $z_{\text{low}}^{(4)} = 22.0721$, at index $k = 5$ with $z_{\text{low}}^{(5)} = 22.1342$ and at index $k = 6$ with $z_{\text{low}}^{(6)} = 22.1710$ the fourth has the best lower bound and is chosen next.

$$a = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad (\bar{u}^\ell)^I = \begin{pmatrix} 1.0000 \\ 2.3053 \\ 3.0000 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 10 \\ 3 \end{pmatrix} \qquad \text{(node 5, k=4)}$$

Here, we have to branch upon the second variable, creating nodes at indices $k = 8$ and $k = 9$. At indices 5, 6, 8, 9 we have open problems in the heap of which the fifth gets selected next by the best-bound strategy.

$$a = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \quad (\bar{u}^\ell)^I = \begin{pmatrix} 1.0000 \\ 2.3052 \\ 4.0000 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 10 \\ 4 \end{pmatrix} \qquad \text{(node 6, k=5)}$$

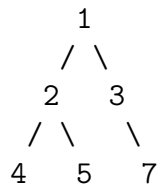
Again we need to branch upon the second variable which results in adding the subproblems $k = 10$ and $k = 11$ to the heap of open problems. From the next node at index $k = 10$ we get

$$a = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \quad (\bar{u}^\ell)^I = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 2 \\ 4 \end{pmatrix} \qquad \text{(node 7, k=10)}$$

whose objective gives the better global $z_{\text{up}} = 22.0315$ which dominates all other open problems (indices 6, 8, 9, 11), i.e. their lower bounds are larger than z_{up} which means solving them could not give a better integer solution than the one we have already found. (Unless their $z_{\text{low}}^{(k)}$ is not actually the lowest value the objective can attain like in those negative estimates D^+ and D^- before. This might occur later in the second example, see Test 3).

No open problems are left, the branch-and-bound algorithm terminates with the POD optimal solution $\bar{u}^\ell$ found.

The branch-and-bound tree looks like this



/
10

treated (due to strong branching they might have already been solved before) in the order 1-3-7-2-4-5-10 where the dominated problems at indices 6, 8, 9 and 11 are not listed.

Hence, solving this mixed-integer optimal control problem with the branch-and-bound algorithm with strong branching and two-phase method required 7 nodes in the branch-and-bound tree and 15 linear-quadratic optimal control problems (LPs) to be solved. Because of the strong branching there are more solved LPs than nodes in the tree: Twice we solved the two child problems for a discarded branching candidate and four open problems were dominated (but previously solved).

5.2.2 Comparison of selection strategies

variable selection	node selection	# nodes	LPs	computational time
strong	two	7	15	20.72 s
	best	5	13	20.77 s
	depth	8	13	18.73 s
pseudo	two	5	7	18.11 s
	best	3	6	16.51 s
	depth	7	9	17.59 s
frac	two	7	7	19.62 s
	best	3	3	14.16 s
	depth	5	5	14.87 s

Table 3: Test 1, POD: This table shows the number of nodes in the branch-and-bound tree, the number of linear-quadratic optimal control problems (LPs) that were solved and the required computational time for each of the methods for selecting the next variable to branch upon and for selecting the next node.

We solve the mixed-integer optimal control problem with model reduction for different variable selection strategies (strong branching, pseudocosts branching and maximum fractional branching, see Section 4.2), and node selection strategies (depth-first, best-bound and two-phase method, see Section 4.3). They all find the same integer-valued solution.
In Table 3 we denoted the resulting number of nodes in the branch-and-bound

tree, the number of linear-quadratic optimal control problems (LPs) and the required computational time for each of the strategies.

Compared to the two-phase method for the **strong branching** strategy which we have discussed in detail in the previous section, we see in the second row of Table 3 that the best-bound selection explores less nodes. In fact, after having solved the root node it selects the node at index $k = 2$ (instead of $k = 3$ as the two-phase method did) and thus gives the solution within only five nodes treated in the order: 1-2-4-5-10. Nevertheless, it is not much faster as it had to solve 13 LPs due to the strong branching and do the high-order optimization in the root node.
Depth-first search treats the nodes in the order 1-3-7-6-2-5-11-10 with 8 nodes in total for which also 13 LPs were solved during the strong branching.

Even though the **pseudocosts branching** solves significantly less LPs it is not much faster in terms of computational time. This can be explained by the fact that a large part is consumed to solve the (high-order) root node whereas the all other relaxations (ROMs) are very small and cheap to solve. A more significant difference might be seen if we actually had nonlinear problems to solve.
The two-phase method treats the nodes in the order 1-3-7-2-4 where 5 and 6 were dominated. It selects a different branching variable in the node at index $k = 2$ where it branches upon the second variable (instead of the third as the strong branching did). So we have $n^+ = n^- = (0, 2, 1)$, i.e. we branched twice on variable u_2^I and once on variable u_3^I. All branching candidates in the strong-branching initialization are counted in $n^+ = n^-$, also if they were discarded.
The best-bound node selection strategy also skips the problems at indices 3 and 7 and finds the optimal solution after treating the nodes 1-2-4 within only three nodes.
For the depth-first strategy we have to treat 7 nodes in the order 1-3-7-6-5-4 and the number of branching decisions is given as $n^+ = n^- = (0, 3, 1)$.

From both $n^+ = n^-$ we can see that we have never branched on the first variable, there it seems clear that we have to use the best insulation. Otherwise, we would lose too much heat since the boundary segment is very long. A further idea could be changing the parametrization of the boundary to make the segments of approximately the same length. The problem should still be manageable with for example 5 boundary segments.

Reliability branching equals the strong branching here as we normally do not reach the treshold $\tau = 5$ when it would switch to the pseudocosts branching strategy.

Maximum fractional branching is said to perform poorly and about as well as randomly choosing a variable in [1]. However, Table 3 presents a different view – we might conclude that it can in fact be suitable for problems where only very few variables are required to be integers and hence there are not so many branching candidates and not so many possibly bad decisions to make.

5.3 Test 2: May 3rd, n = 4

The outside temperature will stay the same as in Test 1, depicted in Figure 2, but now we choose $n = 4$ same-sized heating tiles with $\alpha_i^C = 0.075$ such that the heating costs also stay the same as in the first test (we did not explicitly consider the size of the heating tile in the cost functional). The heating tiles are arranged in counterclockwise order: u_1^C is acting on $\Omega_1 = [0, 0.5]^2$, u_2^C is acting on $\Omega_2 = [0.5, 1] \times [0, 0.5]$, u_3^C on $\Omega_3 = [0.5, 1]^2$ and u_4^C is acting on $\Omega_4 = [0, 0.5] \times [0.5, 1]$.

All branching variable selection methods and node selection methods give the same optimal solution with

$$\bar{u}^I = \bar{u}^{\ell I} = \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}$$

which corresponds to a cheaper insulation than in the first test, where we chose a better insulation for the second boundary segment, the window. This is due to the fact that we can heat more specifically in the areas where it is colder without over-heating in the area where the inner walls meet and no heat transition happens.
Figure 12 shows that in the lower-left corner where the inner walls meet we have to heat the least. In the lower-right corner we have to heat a little more as there is a part of an outer wall, but with a very good insulation. In the upper-right corner we have to heat more, but the most in the upper-left corner where we use a cheaper insulation for the outer wall.

Comparing the temperatures at certain points in the room for the current four-tile case (Figure 13) and the previous one-tile case (Figure 5) we may observe that in the lower-left corner we do now (Test 2) not overshoot the desired temperature as much as before (Test 1) and that due to the cheaper

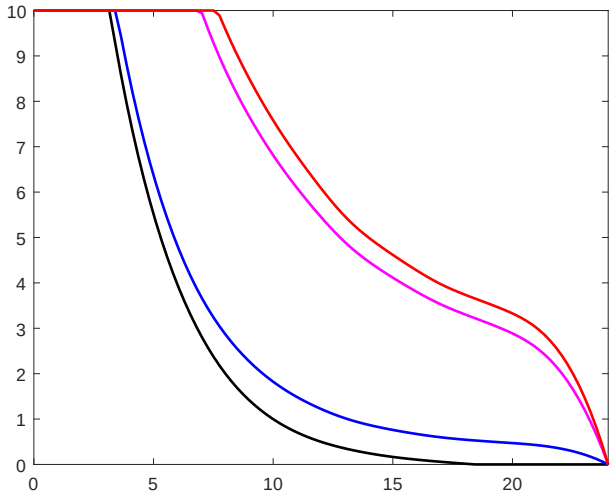


Figure 12: Test 2: Distributed and time-dependent part of the POD optimal control $\bar{u}^{\ell C}$ during the day (in hours). Lower-left (black), lower-right (blue), upper-right (pink) and upper-left heating tile (red).

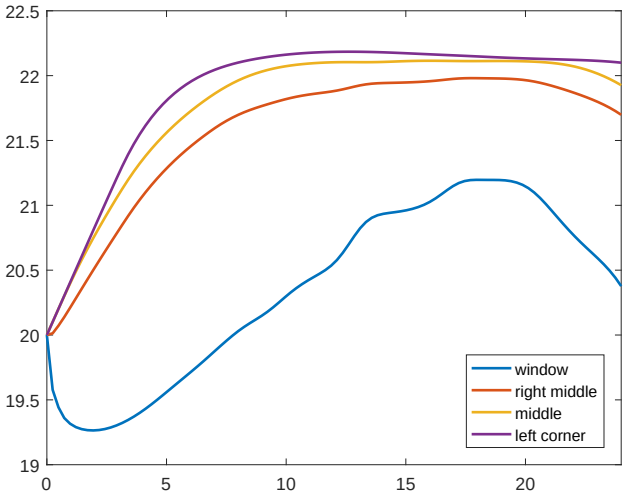


Figure 13: Test 2: Temperature $y(x, \cdot)$ for POD optimal control $\bar{u}^{\ell}$ at certain points in the room: near the window ($x = (0.5222, 1)$, blue), in the middle of the right outer wall ($x = (1, 0.5)$, red), approximately in the middle of the room ($x = (0.5082, 0.4964)$, yellow) and in the lower-left corner ($x = (0, 0)$, purple)

insulation for the window it is now significantly colder at the point near the window. With the better insulation in Test 1 the temperature at the window did not fall below 19.5°C in the beginning and stay only slightly below 21°C at the end of the day, see Figure 5.

variable selection	node selection	# nodes	LPs	computational time
strong	two	6	13	18.72 s
	best	4	11	18.34 s
	depth	9	15	19.88 s
pseudo	two	5	7	16.92 s
	best	3	6	16.99 s
	depth	8	10	17.48 s
frac	two	4	4	15.63 s
	best	3	3	15.01 s
	depth	4	4	17.11 s

Table 4: Test 2, POD: This table shows the number of nodes in the branch-and-bound tree, the number of linear-quadratic optimal control problems (LPs) that were solved and the required computational time for each of the methods for selecting the next variable to branch upon and for selecting the next node.

The rank $\ell = 5$ POD basis has an orthogonality error $e^{\Psi} = 9.1264 \cdot 10^{-13}$ and an energy ratio of $\mathcal{E}(5) = 1 - 8.3029 \cdot 10^{-6}$. The running times and required nodes in the tree as well as the number of solved relaxations can be seen in Table 4.

For strong branching and the two-phase method we show the actual error compared to the a-posteriori estimate in Figure 14 and the difference in the objective in Table 5. The POD basis elements look very similar to the ones in Test 1, they as well show the structure of the room.

For pseudocosts branching and the best-bound strategy we get $n^+ = (0, 1, 1)$ and $n^- = (0, 1, 2)$. So, again it is obvious that we do not have to branch upon the first variable, but choose the best available insulation there. Further, we se that we solved only one up-child which arose from branching on the third variable, but two from branching down on the third variable. This is because we found the integer solution in the down-child with index 4 (giving a global upper bound) and pruned the up-child with index 5 due to its larger lower bound.

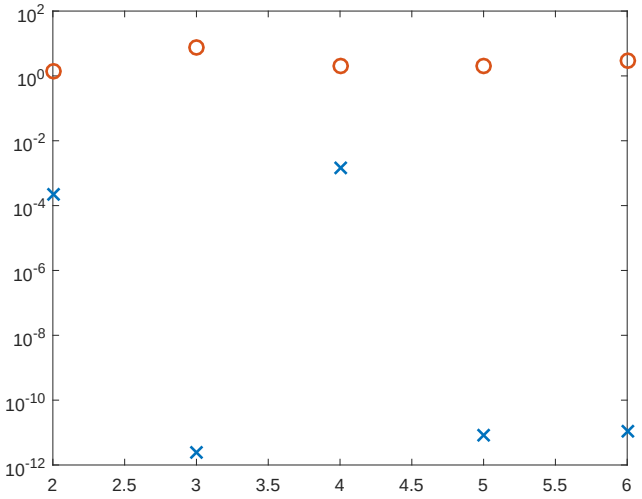


Figure 14: Test 2: Error between optimal $\bar{u}_{ab}$ and $\bar{u}_{ab}^{\ell}$ in the U-norm (blue crosses) and a-posteriori estimator (red circles) at each of the explored nodes in the branch-and-bound algorithm performed by strong branching and the two-phase node selection strategy.

node	$\lvert\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}_{ab}^{\ell})\rvert$
2	0.0175
3	3.2467e-08
4	0.0440
5	1.9037e-06
6	7.9034e-07

Table 5: Test 2: Difference in the cost function values for the optimal and POD optimal control, i.e. $\lvert\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}_{ab}^{\ell})\rvert$ at each node 2-6 on the order they were treated according to the two-phase node selection method, variable selection by strong branching.

5.4 Test 3: June 4th, n = 1

We consider another sample of weather data from a warmer day in the beginning of June in Constance as depicted in Figure 15. The measurements were again taken from [8]. The average day temperature is 18.0°C.

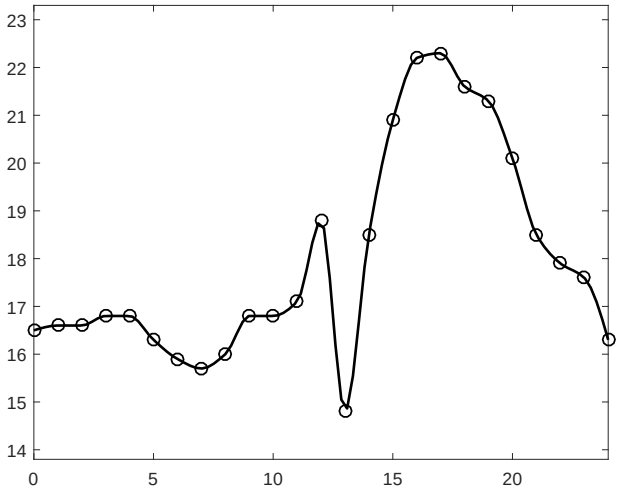


Figure 15: Test 3 and 4: Hourly temperature measurements (circles) and linear interpolation y_a from midnight (0h) till midnight (24h) on June 4th, 2017 in Constance, Germany

We consider again $n = 1$ underfloor heating tile acting on the whole domain Ω and perform the time-stepping by the implicit Euler method. The Rannacher Smoothing does not give as smooth results for the distributed controls (we will show this in Remark 11). We set $\alpha^C = 0.3$ as in Test 1 and $\alpha_Q = 20$. To gain some insights we run the branch-and-bound method without any model reduction for the different selection strategies, see Table 6.

Here we notice the trouble we were talking about in Section 2.2: we might have different controls giving quite similar results. Apparently the warmer outside temperature has made the choice for the third variable less clear. Depending on which methods we chose we get three different optimal insulations for the shortest boundary segment Γ_3. From looking at the objective we might decide that $\bar{u}_3^I = 8$ is the best choice.
There are a few things that could have influenced this. First, we are not in exact arithmetics but solve the relaxations up to certain tolerances. Unfortunately, lowering for instance the tolerance for the relative cost functional increment from 10^{-5} to 10^{-7} only yields a lot more iteration steps as well as

variable s.	node s.	#nodes	LPs	comp. time	$\bar{u}^I$	$\hat{J}^h(\bar{u})$
strong	two	8	25	203.28 s	[3,7,7]	14.7568
	best	5	19	142.98 s	[3,7,7]	14.7568
	depth	12	29	193.39 s	[3,7,9]	14.7348
pseudo	two	7	11	66.66 s	[3,7,8]	14.6762
	best	4	9	60.38 s	[3,7,8]	14.6762
	depth	13	17	119.45 s	[3,7,9]	14.7348
frac	two	7	7	45.67 s	[3,7,8]	14.6762
	best	4	4	28.09 s	[3,7,8]	14.6762
	depth	13	13	141.88 s	[3,7,9]	14.7348

Table 6: Test 3, full: This table shows the number of nodes in the branch-and-bound tree, the number of linear-quadratic optimal control problems (LPs) that were solved and the required computational time for each of the methods for selecting the next variable to branch upon and for selecting the next node. The last two columns show the optimal integer solution found and the objective value.

increasing the maximal number of iterations in the optimization loop does. But that means there could still be a slightly better solution, for example we might find a solution with a lower objective in a child node which results in negative values in the estimators D^+ or D^-. Second, we cannot provide uniqueness of an optimal solution.

In Figure 16 which shows the distributed controls for the optimal boundary controls we can observe what we were thinking of in Section 2.2. A cheaper insulation $\bar{u}^I = [3, 7, 9]$ goes together with a bit more heating (blue) and gives similar temperatures and objectives as a better insulation $\bar{u}^I = [3, 7, 8]$ alongside a bit less heating (red). The difference in the distributed controls is depicted in Figure 17 to get a better quantitative impression.
The choice $\bar{u}^I = [3, 7, 7]$ yields only minimally less heating than $\bar{u}^I = [3, 7, 8]$, the maximal deviation is 0.0053, so the curve would look identical to the red one in Figure 16. So, because of having nearly the same $\bar{u}^C$ this could result in the a bit more expensive insulation giving the slightly larger cost function value for the choice $\bar{u}^I = [3, 7, 7]$.

Nevertheless, from a full optimization of the root node we compute a rank $\ell = 5$ POD basis Ψ which has an error in orthogonality of $e^\Psi = 5.7372 \cdot 10^{-13}$ and an energy ratio of $\mathcal{E}(5) = 1 - 3.1265 \cdot 10^{-7}$.

Comparing the computational times in Table 6 and 7 we get a speed-up of factor 2 to 9 when using the POD method, in detail 5.7 to 9.3 for strong

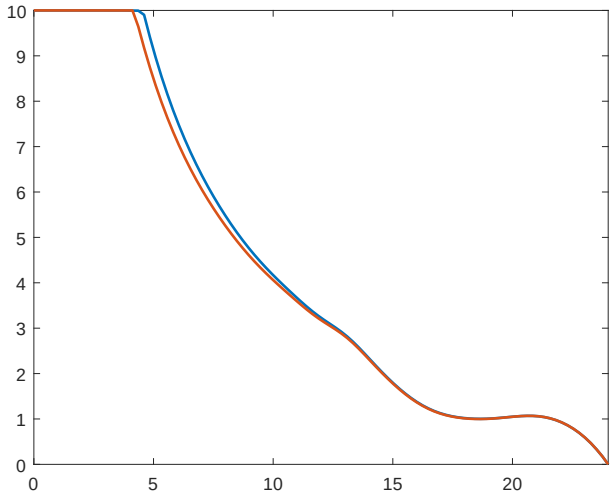


Figure 16: Test 3: Optimal distributed controls with respect to time: for $\bar{u}^I = [3, 7, 8]$ (red) and $\bar{u}^I = [3, 7, 9]$ (blue).

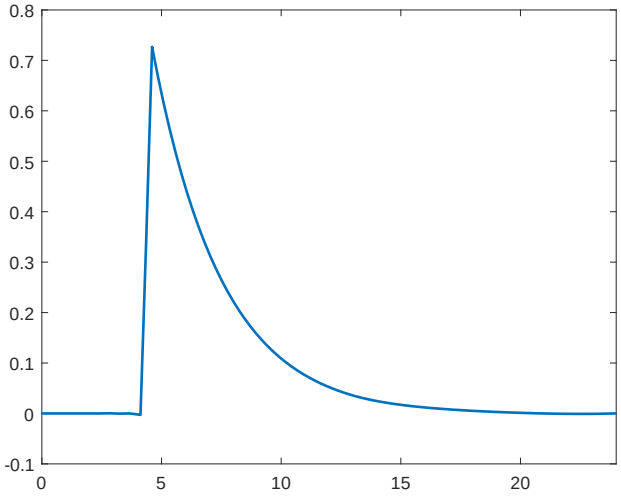


Figure 17: Test 3: Difference in the distributed controls with respect to time for slightly worse $\bar{u}_3^I = 9$ insulation minus slightly better $\bar{u}_3^I = 8$ insulation.

variable s.	node s.	#nodes	LPs	comp. time	$\bar{u}^{\ell I}$	$\hat{J}^h(\bar{u}^\ell)$
strong	two	6	15	21.87 s	[3,7,9]	14.8623
	best	5	19	24.45 s	[3,7,7]	14.7569
	depth	12	19	22.97 s	[3,7,9]	14.8623
pseudo	two	7	11	19.55 s	[3,7,8]	14.6746
	best	4	9	17.77 s	[3,7,8]	14.6762
	depth	9	14	20.72 s	[3,7,9]	14.8623
frac	two	7	7	16.71 s	[3,7,8]	14.6762
	best	4	4	15.69 s	[3,7,8]	14.6762
	depth	16	16	20.78 s	[3,7,8]	14.6762

Table 7: Test 3, POD: This table shows the number of nodes in the branch-and-bound tree, the number of linear-quadratic optimal control problems (LPs) that were solved and the required computational time for each of the methods for selecting the next variable to branch upon and for selecting the next node. The last two columns show the optimal integer solution found and the objective $\hat{J}^h$ at the POD optimal solution.

branching, 3.4 to 5.8 for pseudocosts branching and 1.8 to 6.8 for the maximum fractional branching for these comparably small problems already.

If we increase $\ell = 15$ also the strong-two and fractional-depth combinations yield the same results as their full variants, but then the POD basis already has quite a large error in orthogonality $e^\Psi = 0.99633$. Moreover, from Table 7 we choose the lowest objective which was found for the pseudocosts branching and two-phase node-selection strategies for which we will have a look at the resulting temperatures in the room and the a-posteriori error estimates.

The temperatures near the window and the right outer wall reflect the sudden drop in temperature at hour 13, see Figure 18. The a-posteriori error estimator again overshoots the actual error as displayed in Figure 19 and the difference in the objectives can be seen in Table 8, both for the pseudocosts branching and two-phase method. We solved $n^+ = n^- = (2, 2, 1)$ child-problems whereas for the best-bound search we get $n^+ = (1, 1, 1)$ and $n^- = (2, 2, 1)$, so two up-childs were pruned instead of solved.

Remark 9. The choice of the parameter α_Q influences the coercitivity of the cost functional. One can think of this when dividing $\hat{J}$ by α_Q and computing the coercitivity constant as in (2.29) again. Then it becomes apparent that smaller values of α_Q increase the coercitivity of $\hat{J}$. Thus, if we choose $\alpha_Q = 10$

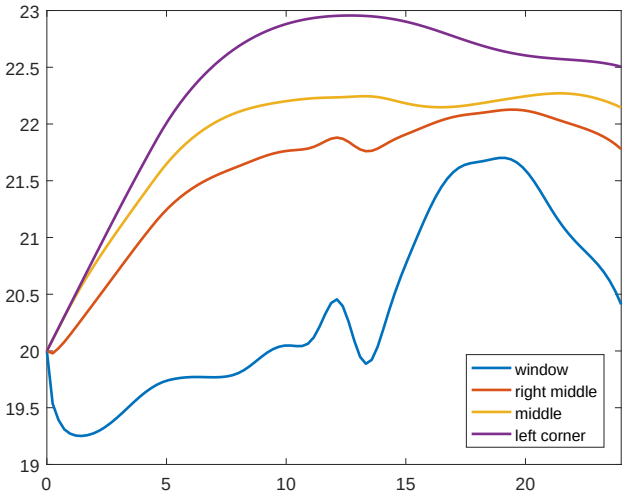


Figure 18: Test 3: Temperature $y(x, \cdot)$ for optimal control $\bar{u}^\ell$ at certain points in the room: near the window ($x = (0.5222, 1)$, blue), in the middle of the right outer wall ($x = (1, 0.5)$, red), approximately in the middle of the room ($x = (0.5082, 0.4964)$, yellow) and in the lower-left corner ($x = (0, 0)$, purple). The branch-and-bound algorithm was performed by the pseudocosts branching and the two-phase selection strategies.

node	$\lvert\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}^\ell_{ab})\rvert$
2	0.1716
3	2.9432e-05
4	1.3794e-07
5	0.0803
6	7.6155e-06
7	4.0775e-08

Table 8: Test 3: Difference in the cost function values for the optimal and POD optimal control, i.e. $|\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}^\ell_{ab})|$ at each node 2-7 on the order they were treated according to the two-phase node selection method, variable selection by pseudocosts branching.

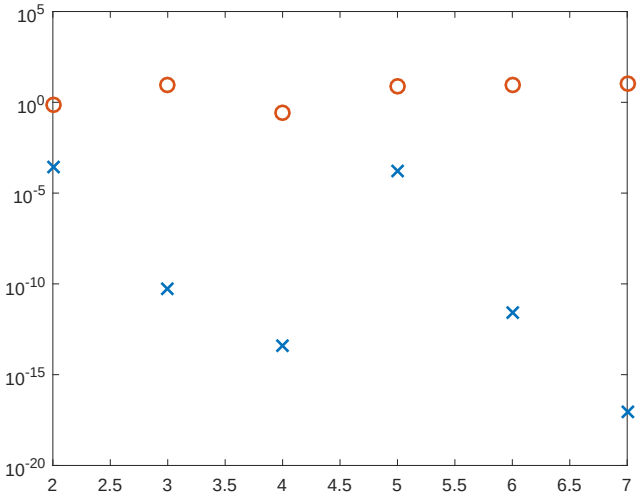


Figure 19: Test 3: Error between high- and low-dimensional optimal solution $\|\bar{u}_{ab} - \bar{u}^{\ell}_{ab}\|_U$ (blue crosses) and a-posteriori estimator (red circles) at each node in the order they were treated. The branch-and-bound algorithm was performed by the pseudocosts branching and the two-phase selection strategies.

instead of the previously $\alpha_Q = 20$, we get for all strategies the same (unique) optimal integer solution $\bar{u}^I = \bar{u}^{\ell I} = [5, 8, 8]$. ◊

5.5 Test 4: June 4th, n = 4

Let us consider this example with a four-tile underfloor heating $n = 4$ for the weather data as in Figure 15. We choose again $\alpha_Q = 20$, $\alpha_i^C = 0.075$ for $1 \leq i \leq 4$ and the implicit Euler method for solving the ODEs.

The error in orthogonality for the rank $\ell = 5$ POD basis is $e^{\Psi} = 7.9193 \cdot 10^{-14}$ and the energy ratio $\mathcal{E}(5) = 1 - 7.8993 \cdot 10^{-6}$. Table 9 shows even better than in the third test that an optimal solution does not have to be necessarily unique as the difference in the objective is pretty small: $6.5 \cdot 10^{-4}$. Running the branch-and-bound algorithm with full optimizations in all nodes yields the same results as for the POD-based approach, except for the strong branching and two-phase method that also finds the slightly better $\bar{u}^I = [4, 8, 8]$ within 158.58 s, so it takes up 6 times as much computational time.

Comparing the temperatures at certain points in the room for the four-tile case (Figure 21) to the one-tile case (Figure 18) we can see that with more heating tiles we can heat more specifically and thus the temperatures within the room differ less.

variable s.	node s.	#nodes	LPs	comp. time	$\bar{u}^{\ell I}$	$\hat{J}^h(\bar{u}^\ell)$
strong	two	7	17	25.72 s	[4,7,9]	13.7600
	best	5	15	27.47 s	[4,8,8]	13.7594
	depth	12	19	26.37 s	[4,7,9]	13.7600
pseudo	two	9	13	27.11 s	[4,8,8]	13.7594
	best	6	11	24.71 s	[4,8,8]	13.7594
	depth	12	17	25.22 s	[4,7,9]	13.7600
frac	two	9	9	24.34 s	[4,8,8]	13.7594
	best	6	6	22.30 s	[4,8,8]	13.7594
	depth	12	12	23.70 s	[4,7,9]	13.7600

Table 9: Test 4, POD: This table shows the number of nodes in the branch-and-bound tree, the number of linear-quadratic optimal control problems (LPs) that were solved and the required computational time for each of the methods for selecting the next variable to branch upon and for selecting the next node. The last two columns show the optimal integer solution found and the objective $\hat{J}^h$ at the POD optimal solution.

node	$\lvert\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}^\ell_{ab})\rvert$
2	2.9152e-06
3	1.8152e-06
4	1.0621e-05
5	9.3107e-07
6	8.0439e-07

Table 10: Test 4: Difference in the cost function values for the optimal and POD optimal control, i.e. $|\hat{J}^h(\bar{u}_{ab}) - \hat{J}^h(\bar{u}^\ell_{ab})|$ at each node 2-6 on the order they were treated according to the best-bound node selection method, variable selection by pseudocosts branching.

For pseudocosts branching and the best-bound selection we solve $n^+ = n^- = (1, 2, 1)$ child nodes, the a-posteriori error estimator can be seen in Figure 22 and the difference in the cost functional in Table 10 for each of the treated nodes.

Remark 10. Like we have explained in the Remark 9, lowering α_Q will make the solution more unique. Setting $\alpha_Q = 10$ will give for all strategies the same

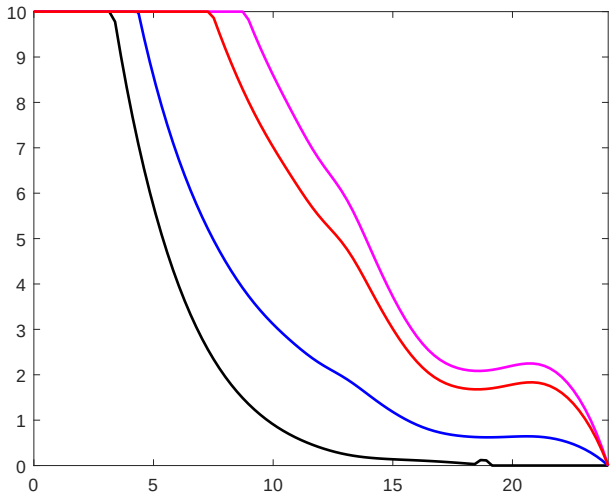


Figure 20: Test 4: Distributed and time-dependent part of the POD optimal control $\bar{u}^{\ell C}$ during the day (in hours). Lower-left (black), lower-right (blue), upper-right (pink) and upper-left heating tile (red). The branch-and-bound algorithm was performed by the pseudocosts branching and the best-bound selection strategies.

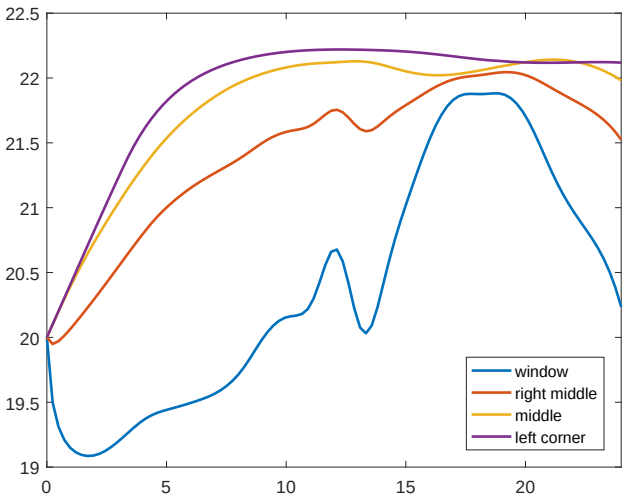


Figure 21: Test 4: Temperature $y(x, \cdot)$ for optimal control $\bar{u}^{\ell}$ with pseudocosts branching and best-bound search at certain points in the room: near the window ($x = (0.5222, 1)$, blue), in the middle of the right outer wall ($x = (1, 0.5)$, red), approximately in the middle of the room ($x = (0.5082, 0.4964)$, yellow) and in the lower-left corner ($x = (0, 0)$, purple).

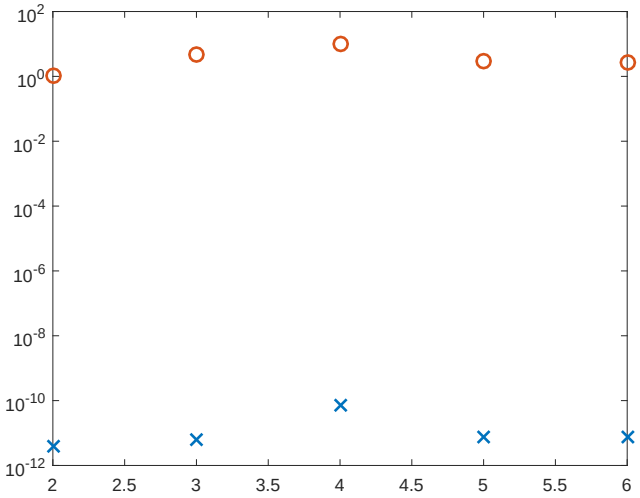


Figure 22: Test 4: Error between high- and low-dimensional optimal solution $\|\bar{u}_{ab} - \bar{u}^{\ell}_{ab}\|_U$ (blue crosses) and a-posteriori estimator (red circles) at each node in the order they were treated. The branch-and-bound algorithm was performed by the pseudocosts branching and the best-bound selection strategies.

$\bar{u}^{\ell I} = [5, 8, 8]$, this is a cheaper insulation on Γ_1 because we do not punish deviations from the desired temperature that much anymore so colder temperatures are more acceptable. ◊

Remark 11. As we have mentioned we did not choose to solve the ODEs by Rannacher Smoothing for Test 3 and 4. This is motivated by this solution for the distributed controls in Figure 23 for pseudocosts branching with the best-bound node selection strategy (which happens for all other methods as well). Perhaps it struggles with the stronger varying outside temperatures than we had in Test 1 and 2. But also for the implicit Euler method, see Figure 20, the lower-left heating tile (black) has a little bump at hour 18. We could also try to avoid this by punishing rapid changes in the distributed controls in the cost functional, too. ◊

Remark 12. Increasing the cost for the heating, setting $\alpha^C_i = 0.3$ (instead of $\alpha^C_i = 0.075$ as we had in Test 4) for $1 \leq i \leq 4$ makes heating four times as expensive. Then the optimal control consists of less heating (see Figure 24) and a better insulation $\bar{u}^I = [1, 6, 7]$. This also matches our expectations from real life. ◊

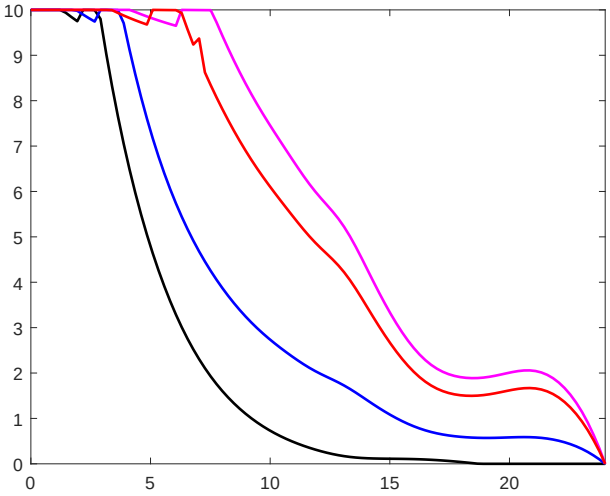


Figure 23: Test 4 with Rannacher Smoothing: $\bar{u}^{\ell C}$ by the pseudocosts branching and the best-bound search during the day (in hours) with lower-left (black), lower-right (blue), upper-right (pink) and upper-left heating tile (red).

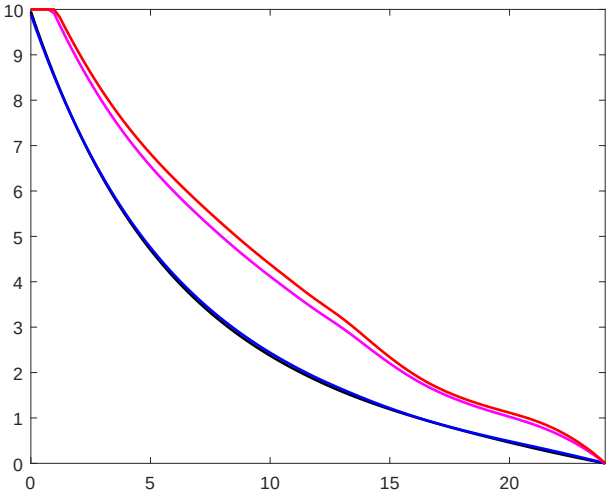


Figure 24: Test 4 with increased α^C: $\bar{u}^{\ell C}$ during the day (in hours) for lower-left (black), lower-right (blue), upper-right (pink) and upper-left heating tile (red) with pseudocosts branching and best-bound search. The time-stepping was again performed by the implicit Euler method.

6 Conclusion and outlook

In this thesis we presented the branch-and-bound algorithm for solving a linear-quadratic mixed-integer optimal control problem (MIOCP) motivated by the physical task of maintaining a desired temperature in a room and choosing suitable insulation materials for outer walls where heat transition happens. The state variable, i.e. the temperature distribution, was governed by the linear heat equation.
The system had two types of control variables: distributed ones representing individual underfloor heating tiles whereas boundary controls denoted certain choices of insulation material. The first ones were acting on (a subdomain of) Ω and allowed to be continuous and time-dependent whereas the latter had to be chosen from a set of integer values.

The branch-and-bound algorithm required solving of many relaxations of the MIOCP with varying lower and upper bounds on the boundary controls giving a tree-like structure. In fact, these relaxations were standard linear-quadratic optimal control problems which can be solved numerically by the finite element method. In the root node, the relaxation of the original MIOCP, we chose linear Lagrange elements which was also referred to as the high-dimensional model. From the state and adjoint corresponding to the optimal solution we computed a POD basis by solving an eigenvalue problem. Then we continued to solve the further relaxations arising from branching by a reduced-order model with the POD Galerkin ansatz. This provided a significant speed-up of factor 2 to 9 while keeping the error very low.

We want to emphasize that this speed-up happened for a comparably small problem with only three integer boundary controls and maximal four distributed controls. Without any doubt we can conclude that the model reduction using POD will be inevitable for more complex problems and it will be very interesting to see how the presented variable and node selection strategies will perform then. One could think of a larger number of integer controls or actually nonlinear problems arising from more detailed models for describing the temperature, e.g. considering an advection term in the state equation, both increasing the complexity of the MIOCP.

Moreover, we discussed an a-posteriori error estimate to control the error between the low- and highdimensional solution. In all experiments the estimator bounded the real error from above, even though it was not a very sharp estimate. Usually, the a-posteriori error estimator overshot the actual error by several orders of magnitude. One clearly has to dig deeper into a-posteriori error estimation theory to hopefully find a better estimate. Of interest could as well be the spatial distribution of the error, one might imagine a larger error in vibrant areas for instance near the window and consider mesh adaptation.

All in all, we have seen in several tests with varying parameters that the computed solutions match our physical and economical expectations and identified parameters critical to the coercitivity (and uniqueness) of the problem.
If one strongly insisted on keeping the desired temperature as closely as possible we have seen that there might be several similarly optimal solutions. Then, instead of weighting the contrary control tasks against each other one could eventually try multi-objective methods, which can also be seen as a way of increasing the complexity of the problem.

The programming work was done with the background thought of giving the cost functional in augmented Lagrange form such that additional inequalities, e.g. from cutting plane techniques to tighten the integer set before solving a node which can significantly reduce the size of the tree (this known as branch-and-cut algorithm, see [1]), can be incorporated easily.

So definitely, mixed-integer optimal control problems still provide a lot of interesting things to investigate in the future.

Bibliography

[1] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013. 1, 4.2, 4.2, 5.2.1, 5.2.2, 6

[2] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I.* Springer-Verlag, Berlin, 2000. 2.1

[3] R. Denk and R. Racke. *Kompendium der Analysis. Band 2: Maß- und Integrationstheorie, Funktionentheorie, Funktionalanalysis, Partielle Differentialgleichungen.* Vieweg+Teubner, Wiesbaden, 2012. 2.1

[4] L. Evans. *Partial Differential Equations*, volume 19. American Math. Society, Providence, Rhode Island, second edition edition, 2010. 2.1

[5] M. Gubisch and S. Volkwein. POD for Linear-Quadratic Optimal Control, 2013. URL `http://nbn-resolving.de/urn:nbn:de:bsz:352-250378`. 2.3, 2.3, 3, 3.1, 3.2, 3.2, 3.2.1

[6] M. Gubisch and S. Volkwein. Proper orthogonal decomposition for linear-quadratic optimal control. *To appear in P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), Model Reduction and Approximation: Theory and Algorithms. SIAM, Philadelphia, PA*, 2017. 3.1

[7] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Application*, 39:319–345, 2008. 3.1

[8] J. Kachelmann. Messwerte & Klimadaten mit Archiv ab dem Jahr 1781, latest access on July 5th, 2017. URL `https://kachelmannwetter.com/de/messwerte/baden-wuerttemberg/temperatur/20170705-1200z.html`. 5.2, 5.4

[9] The MathWorks, Inc. MATLAB documentation, `eigs`. URL `https://de.mathworks.com/help/matlab/ref/eigs.html#bu2_xjs-5`. 5.1

[10] F. Tröltzsch. *Optimal Control of Partial Differential Equations*, volume 112. American Mathematical Society, 2010. 2, 2.1, 2.1, 2.2, 2.2, 2.3, 2.3

[11] F. Tröltzsch and S. Volkwein. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44:83–115, 2009. 2.5

Acknowledgements

Foremost, I wish to thank my supervisor Prof. Dr. Stefan Volkwein for his inspiring enthusiasm about numerical mathematics, for his great support and for his confidence in my independent work. Whenever I left his office it was with lots of new ideas, helpful suggestions and plenty of motivation. Further, I would like to thank Jun.-Prof. Dr. Gabriele Ciaramella for the interest in my work and taking second review.

Also, I appreciated the opportunity to be a part of Prof. Volkwein's working group. The joint programming work with Dennis Beermann and Stefan Banholzer in the beginning of this thesis was fruitful and fun. In addition, I want to thank Dennis Beermann for his time and advice whenever I had questions; and Stefan Banholzer for the helpful feedback in my writing process. Moreover, I would like to thank Jianjie Lu who introduced me to mixed-integer problems in the course of a numerical seminar and Stefan Trenz for this well-documented optimization codes.

Personally, I would like to thank my family and friends for their love and patience, my fabulous band mate for the huge support and continuous encouragement and all the great adventures we experience and Kitty, my favourite cat who is a loyal companion.