Master’s thesis

FIRST-ORDER
AUGMENTED LAGRANGIAN METHODS
FOR STATE CONSTRAINT PROBLEMS

submitted by
Jonas Siegfried Jehle

at the Faculty of Sciences Department of Mathematics and Statistics

University of Konstanz

and the Department of Mathematics

German Supervisor and 1st Reviewer: Prof. Dr. Stefan Volkwein,
University of Konstanz

Chinese Supervisor and 2nd Reviewer: Prof. Ph.D. Jinyan Fan,
Shanghai Jiao Tong University

Konstanz, 6 September 2018
I hereby declare that the submitted thesis

*First-order augmented Lagrangian methods for state constraint problems*

is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here.

This paper was not previously presented to another examination board and has not been published.

Konstanz, 6 September 2018

.................................
Jonas Siegfried Jehle
Abstract

In this thesis, the augmented Lagrangian method is used to solve optimal boundary control problems. The considered optimal control problem appears in energy efficient building operation and consists of a heat equation with convection along with bilateral control and state constraints. The goal is to fit the temperature (state) to a given prescribed temperature profile covering as few heating (controlling) costs as possible. Numerically, a first-order method is applied to solve the minimization problem occurring within the augmented Lagrangian algorithm. Thereto, we set up and solve the adjoint equation. Both partial differential equations, the state and the adjoint equation, are treated with the finite element Galerkin ansatz combined with an implicit Euler scheme in time. At the end, numerical tests of the created first-order augmented Lagrangian method illustrate the efficiency of the proposed strategy.
### Contents

1 Introduction  
2 Preliminaries  
3 The Augmented Lagrangian Method  
  3.1 The Gradient and the Hessian Matrix  
  3.2 The Advantage of the Expansion  
  3.3 Numerical Approach  
  3.4 Problems with Inequality Constraints  
4 The Convection-Diffusion Model  
  4.1 Weak Formulation  
  4.2 Well-Posedness and Unique Existence  
5 The Optimal Control Problem of the Convection-Diffusion Model  
  5.1 The Augmented Lagrange Function of the Optimal Control Problem  
  5.2 The Derivative of the Augmented Lagrange Function  
  5.3 The Adjoint Equation  
6 Numerical Consideration of the Optimal Control Problem  
  6.1 Solving the State and the Adjoint Equation  
    6.1.1 Galerkin Ansatz and Implicit Euler Method for the State Equation  
    6.1.2 Galerkin Ansatz and Implicit Euler Method for the Adjoint Equation  
  6.2 The Practicable Code  
  6.3 Tests  
    6.3.1 Test 1  
    6.3.2 Test 2  
    6.3.3 Test 3  
7 Conclusion and Outlook  
References
Chapter 1

INTRODUCTION

Starting in the year 1696, when Johann Bernoulli released the established Brachystochrone problem (see [Str69, pages 391-399]), optimal control problems grew to an independent, relevant topic in the field of mathematics - but not alone in theory. After the invention of computers, calculating the solutions of optimal control problems became faster and more efficient. Optimal control searches for the answer to the question which control law is the most favourable one for a given system in such a way that a certain optimality criterion is reached. Typically, like constrained optimization problems, a control problem contains a cost function as well as constraints. The cost function thereby possesses the state variable $y$ and the control variable $u$ as function arguments. The control $u$ may be arbitrarily chosen within the given constraints while the state $y$ is uniquely identified in terms of $u$. The aim is to determine $u$ such that the cost function is minimized and the constraints are fulfilled. Such controls are called optimal. In practice, one application area of optimal control is present in aviation and space technology. The example of the rocket car in the textbook [MS82, pages 10-15] by Macki and Strauss shows why: as often as an airplane is supposed to trail a flight path, issues of optimization are needed to bring into effect. Furthermore, optimal control is employed in medicine, robotics, movement sequences in sports and engineering to name just a few.

Since the augmented Lagrangian method belongs to the class of algorithms for solving constrained optimization problems, it can be used for a numerical approach of an optimal control problem. In 1969, Magnus Hestenes and Michael Powell mentioned the method in their works (see [Hes69, pages 303-320] and [Pow69, pages 283–298]) for the first time. In practice, the algorithm resembles a penalty method. Instead of solving the constrained optimization problem, it is replaced by a series of unconstrained problems where a term including the Lagrange multiplier is being added to the objective as well as a penalty term. The Lagrange multipliers are auxiliary variables supporting the characterization of optimal solutions on the one hand. On the other hand they carry useful sensitivity information. The term of augmentation originates from the penalty term since it distinguishes the common Lagrangian method from the augmented one. It is worth noting that the mathematicians Ralph Tyrrell Rockafellar and Dimitri Bertsekas worked with the augmented Lagrangian method - Rockafellar in relation to convex pogramming in several papers (see [Roc73, pages 555—562], [Roc74, pages 268—285], [Roc76, pages 97—116]) while Bertsekas mentions it repeatedly in his book and extends it to the exponential method of multipliers (see [Ber96]).

In this thesis, we study an optimal boundary control problem appearing in energy efficient building operation. To produce the temperature changes in a room representing the state of the optimal control, we observe a convection-diffusion model which is a parabolic partial differential equation with Robin boundary condition. The room temperature has to be kept inside a certain region controlled by heaters placed on the boundary of the room. The power of the heaters is restricted due to the real life practice and so
they underlie bilateral control constraints. The aim is to track a prescribed temperature profile and minimize the upcoming heating costs while not transgressing the selected state and control constraints, respectively.

In order to achieve our goal, the thesis is structured as follows:

• Chapter 2 deals with basic notations and definitions and provides assistance for solving optimization problems and the arising partial differential equations.

• The augmented Lagrangian method is introduced in Chapter 3, together with examples and important properties.

• In Chapter 4, a weak formulation of the employed convection-diffusion model is given and its well-posedness is proven.

• Chapter 5 is devoted to the introduction of the cost functional and the Lagrange function whose gradient is determined. To work with the gradient, we introduce the adjoint equation by which we can reformulate the gradient in a suitable way.

• The numerical realization of the theoretical work is discussed in Chapter 6. We solve the state and adjoint equation using the finite element Galerkin ansatz combined with an implicit Euler scheme in time. Within the augmented Lagrangian method we use a gradient projection method to perform the minimization.

• Lastly, Chapter 7 is a summary of the thesis and the conclusion of the obtained results. It also presents a brief outlook on subsequent questions and possible modifications.
In this chapter, we provide basic definitions, notations and results, which are used throughout this thesis. The proofs for the results are omitted and references to the literature, where these topics are discussed in detail, are given instead. First of all, we go through some common expressions. Let \( m, n \in \mathbb{N} \) in this chapter.

**Notation 2.1.** Let \( x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{R}^n \) and \( \Omega \subset \mathbb{R}^n \).

- We write
  
  \[ x \leq y : \iff \quad x_i \leq y_i \text{ for } i \in \{1, \ldots, n\} \]
  
  \[ x < y : \iff \quad x_i < y_i \text{ for } i \in \{1, \ldots, n\} \]

  and on basis of that, we define

  \[ (x, y) := \{ z \in \mathbb{R}^n | x < z < y \} \]
  
  \[ [x, y] := \{ z \in \mathbb{R}^n | x \leq z \leq y \} \]

  for \( x < y \). Analogously, we define \( (x, y] \) and \( [x, y) \).

- The **Euclidean norm** is denoted as

  \[ \|x\| := \|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \]

  with the corresponding inner product

  \[ \langle x, y \rangle := x^T y := \sum_{i=1}^{n} x_i y_i. \]

- \( C^k(X), k \in \mathbb{N} \), is the space of all \( k \)-times continuously differentiable functions mapping from an open, non-empty subset \( X \subset \mathbb{R}^n \) to \( \mathbb{R} \). Furthermore, a \( C^\infty \) function is a function that is differentiable for all degrees of differentiation. \( C^\infty \) functions are also called test functions.

- Let \( X, Y \) be topological vector spaces. Then, we call

  \[ L(X, Y) := \{ T : X \to Y \mid T \text{ is a linear and continuous operator} \} \]

  the space of all linear and continuous operators from \( X \) to \( Y \). In particular, \( X' = L(X, \mathbb{R}) \) is called the dual space of \( X \).

- Let \( \Omega \subset \mathbb{R}^n \) be a measure space with Lebesgue measure. The common \( L^p \) space for \( \Omega \) is denoted as \( L^p(\Omega) \) with \( L^p \)-norm

3
\[ \|f\|_{L^p(\Omega)} := \left( \int_{\Omega} |f(x)|^p \, dx \right)^{\frac{1}{p}} \]

for \( f \in L^p(\Omega) \) (\( 1 \leq p < \infty \)). In particular, for \( f, g \in L^2(\Omega) \), it holds

\[ \langle f, g \rangle_{L^2(\Omega)} = \int_{\Omega} f(x) g(x) \, dx. \]

- The Sobolev space \( H^1(\Omega) \) is the completion of \( \{ u \in C^1(\Omega) : \|u\|_{H^1(\Omega)} < \infty \} \) with

\[ \|u\|_{H^1(\Omega)} := \sqrt{\|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega; \mathbb{R}^n)}^2}. \]

- Let \( U, V \) be Hilbert spaces. Let \( S : U \to V \) be a linear, bounded operator. We call \( S^* : V \to U \) the adjoint if it holds \( \langle Su, v \rangle_V = \langle u, S^*v \rangle_U \) for all \( u \in U, v \in V \).

Next, we provide sufficient and necessary conditions for restricted optimization problems of the form

\text{(ROP)} \quad \min J(x) \quad \text{subject to (s.t.)} \quad x \in \mathbb{R}^n \text{ and } e(x) = 0,

where \( J : \mathbb{R}^n \to \mathbb{R} \) represents the cost function and \( e = (e_1, \ldots, e_m)^T : \mathbb{R}^n \to \mathbb{R}^m \) the equality constraints. The following expressions and statements can also be found in [LY08] and [NW06].

**Definition 2.2.** Let \( x^* \in \mathbb{R}^n \) and \( U(x^*) \subset \mathbb{R}^n \) be a neighbourhood of the point \( x^* \).

- We call \( x^* \) local solution of \( \text{(ROP)} \) if it holds

\[ e(x^*) = 0 \quad \text{and} \quad J(x^*) \leq J(x) \text{ for all } x \in U(x^*) \text{ with } e(x) = 0. \]

- The point \( x^* \) is a strict local solution of \( \text{(ROP)} \) if it holds

\[ e(x^*) = 0 \quad \text{and} \quad J(x^*) < J(x) \text{ for all } x \in U(x^*) \text{ with } e(x) = 0. \]

- We call \( x^* \) a global solution of \( \text{(ROP)} \) if it holds

\[ e(x^*) = 0 \quad \text{and} \quad J(x^*) \leq J(x) \text{ for all } x \in \mathbb{R}^n \text{ with } e(x) = 0. \]

Analogously, the concept of a strict global solution is given.

**Definition 2.3.** Let \( \bar{x} \in \mathbb{R}^n \) fulfill \( e(\bar{x}) = 0 \). Then, the point \( \bar{x} \) is called regular relative to \( e(x) = 0 \) if and only if \( \nabla e_1(\bar{x}), \nabla e_2(\bar{x}), \ldots, \nabla e_m(\bar{x}) \in \mathbb{R}^{1 \times n} \) are linearly independent in \( \mathbb{R}^n \).

**Theorem 2.4** (First-order Necessary Optimality Condition). Let \( x^* \in \mathbb{R}^n \) be a local solution of \( \text{(ROP)} \) and regular. Then, there exists a unique Lagrange multiplier \( \lambda^* \in \mathbb{R}^m \) such that

\[ \nabla J(x^*) + \sum_{i=1}^m \lambda_i^* \nabla e_i(x^*) = \nabla J(x^*) + (\lambda^*)^T \nabla e(x^*) = 0. \]
Proof. A proof can be found in [LY08, Chapter 11] \hfill \square

**Definition and Remark 2.5.** We define the common Lagrange function

\[ L(x, \lambda) := J(x) + \lambda^T e(x) = J(x) + \langle \lambda, e(x) \rangle. \]

Using its gradient, we can rewrite the first-order necessary optimality condition (2.1) and the constraints as

\[
\nabla_x L(x^*, \lambda^*) = \nabla J(x^*) + (\lambda^*)^T \nabla e(x^*) = 0, \\
\nabla_\lambda L(x^*, \lambda^*) = e(x^*)^T = 0,
\]

or just \( \nabla L(x^*, \lambda^*) = 0. \)

**Theorem 2.6** (Second-order Necessary Optimality Condition). Let \( x^* \) be regular and a local minimum of \( J \) under the constraint \( e(x) = 0 \). Then, the \( n \times n \)-matrix

\[
\nabla_{xx} L(x^*, \lambda^*) = \nabla^2 J(x^*) + (\lambda^*)^T \nabla^2 e(x^*) = \nabla^2 J(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 e_i(x^*)
\]

is positive semidefinite on the nullspace of \( \nabla e(x^*) \), i.e.

\[
v^T \nabla_{xx} L(x^*, \lambda^*) v \geq 0 \quad \text{for all } v \in \ker(\nabla e(x^*)) := \{v \in \mathbb{R}^n | \nabla e(x^*) v = 0\},
\]

where \( \lambda^* \) represents the Lagrange multiplier from 2.4.

**Proof.** A proof can be found in [LY08, Chapter 11] \hfill \square

**Theorem 2.7** (Second-order Sufficient Optimality Condition). Let \( x^* \in \mathbb{R}^n \) with \( e(x^*) = 0 \) and let \( \lambda^* \in \mathbb{R}^m \) satisfy

\[
\nabla J(x^*) + (\lambda^*)^T \nabla e(x^*) = 0.
\]

Further, let \( x^* \) be regular and \( \nabla_{xx} L(x^*, \lambda^*) \) be positive definite on the nullspace of \( \nabla e(x^*) \), i.e.

\[
v^T \nabla_{xx} L(x^*, \lambda^*) v > 0 \quad \text{for all } v \in \ker(\nabla e(x^*))\setminus\{0\}.
\]

Then, \( x^* \) is a strict local minimum of \( J \) subject to the constraint \( e(x) = 0 \). This means that \( x^* \) is a strict local solution of \( (ROP) \).

**Proof.** A proof can be found in [LY08, Chapter 11] \hfill \square

At last, assistive equipment for solving a partial differential equations, which occurs in the constraints of our optimal boundary control problem, is allocated.

**Theorem 2.8** (Trace Theorem). Let \( \Omega \subset \mathbb{R}^n \) be bounded with \( C^1 \)-boundary. Then, there exists a bounded linear operator

\[
T : H^1(\Omega) \rightarrow L^2(\partial\Omega)
\]

such that
• $Tu = u|_{\partial \Omega}$ for $u \in H^1(\Omega) \cap C(\overline{\Omega})$,

• $\| Tu \|_{L^2(\partial \Omega)} \leq C \| u \|_{H^1(\Omega)}$ for each $u \in H^1(\Omega)$ with constant $C = C(\Omega)$.

Proof. A proof can be found in [Eva10, pages 258-259].

Definition 2.9. Let $(V, \langle \cdot, \cdot \rangle_V)$ be a separable Hilbert space and let $(A(t))_{t \in [0,T]}$ be a family of operators $A : [0, T] \to L(V, V')$. The corresponding bilinear form $a : [0, T] \times V \times V \to \mathbb{R}$ is defined by

$$a(t; \varphi, \psi) := -\langle A(t)\varphi, \psi \rangle_{V', V} \quad (t \in [0, T], \varphi, \psi \in V).$$

The bilinear form is called coercive, if there exist $\alpha > 0$ and $\beta \geq 0$ with

$$a(t; \varphi, \varphi) \geq \alpha \| \varphi \|^2_V - \beta \| \varphi \|^2_H \quad (t \in [0, T], \varphi \in V).$$

In the case $\beta = 0$ we say $a$ is strict coercive.

Theorem 2.10. Let $(V, \langle \cdot, \cdot \rangle_V)$ and $(H, \langle \cdot, \cdot \rangle_H)$ be separable Hilbert spaces which form a Gelfand triple $V \hookrightarrow H = H' \hookrightarrow V'$. Let $A \in C([0, T], L(V, V'))$ be coercive. Let the Hilbert space $W(0, T) := L^2(0, T; V) \cap H^1(0, T; V')$ be endowed with the inner product

$$\langle \varphi, \psi \rangle_{W(0, T)} := \int_0^T \langle \varphi(t), \psi(t) \rangle_V + \langle \varphi_t(t), \psi_t(t) \rangle_{V'} \, dt \quad (\varphi, \psi \in W(0, T)).$$

Then, for all $f \in L^2(0, T; V')$ and $y_0 \in H$, there exists a unique solution $y \in W(0, T)$ of the abstract evolution equation

$$y_t(t) - A(t)y(t) = f(t) \quad (t \in (0, T)),$$

$$y(0) = y_0.$$

In particular, the mapping

$$W(0, T) \to L^2(0, T; V') \times H,$$

$$y \mapsto (y_t - Ay, y(0))$$

is an isomorphism of Hilbert spaces. Furthermore, the inverse mapping, which is the so called solution mapping, is linear and continuous. Hence, there is a constant $C > 0$ such that

$$\| y \|_{W(0, T)} \leq C \left( \| f \|_{L^2(0, T; V')} + \| y_0 \|_H \right)$$

is true for all $(f, y_0) \in L^2(0, T; V') \times H$, where $y \in W(0, T)$ is the solution of the abstract evolution equation.

Proof. A proof can be found in [Paz92, Chapter 5] or in [Den13, Chapter 8]. The fact that $(W(0, T), \langle \cdot, \cdot \rangle_{W(0, T)})$ is a Hilbert space is proven in [Trö10, pages 146-148].

Remark 2.11. Choosing $V = H^1(\Omega)$ and $H = L^2(\Omega)$ for a domain $\Omega \subset \mathbb{R}^n$, we obtain a Gelfand triple $V \hookrightarrow H \hookrightarrow V'$. For the precise definition of a Gelfand triple see [Trö10, page 147].
The Augmented Lagrangian Method

In this chapter, we introduce the augmented Lagrangian method. First, the method is applied to solve an optimization problem with equality constraints. Properties of the method are shown and the numerical realization is presented. At the end of the chapter, we show how to deal with additional inequality constraints. The chapter is orientated on [Ber99, Chapter 3 and 4]. Let $m, n \in \mathbb{N}$ and $m \leq n$. We consider the restricted optimization problem

\[
\text{(ROP)} \quad \min_{x} J(x) \quad \text{s.t.} \quad x \in \mathbb{R}^n \text{ and } e(x) = 0,
\]

for the cost function $J : \mathbb{R}^n \to \mathbb{R}$ and the equality constraints $e = (e_1, \ldots, e_m)^T : \mathbb{R}^n \to \mathbb{R}^m$.

While we use the Lagrange function

\[ L(x, \lambda) = J(x) + \lambda^T e(x), \]

in the common Lagrangian method with $\lambda \in \mathbb{R}^m$, choosing $c > 0$ and adding the term \[ \frac{c}{2} \|e(x)\|^2 = \frac{c}{2} e(x)^T e(x) \]
gives us the augmented Lagrange function

\[ (3.1) \quad L_c(x, \lambda) = J(x) + \lambda^T e(x) + \frac{c}{2} \|e(x)\|^2. \]

We realize that $L_c(x, \lambda)$ is the common Lagrange function for the problem

\[
\min J(x) + \frac{c}{2} \|e(x)\|^2 \quad \text{s.t.} \quad x \in \mathbb{R}^n \text{ and } e(x) = 0.
\]

As a consequence, this problem has the same minima as (ROP).

3.1 || The Gradient and the Hessian Matrix

Now, we calculate the first and second derivative of the augmented Lagrange function (3.1). For the gradient we obtain

\[
\nabla L_c(x, \lambda) = \left( \nabla_x L_c(x, \lambda), e(x)^T \right) \in \mathbb{R}^{1 \times (n+m)},
\]

where the product rule yields

\[
\nabla_x L_c(x, \lambda) = \nabla J(x) + (\lambda + ce(x))^T \nabla e(x)
\]

\[ = \nabla_x L(x, \hat{\lambda}) \in \mathbb{R}^{1 \times n}. \]

We write the Hessian matrix in block matrix form, namely

\[
\nabla^2 L_c(x, \lambda) = \begin{pmatrix}
\nabla^2_{xx} L_c(x, \lambda) & \nabla e(x)^T \\
\n\nabla e(x) & 0
\end{pmatrix} \in \mathbb{R}^{(n+m) \times (n+m)},
\]

where we use the product rule again to get

\[
\nabla^2_{xx} L_c(x, \lambda) = \nabla^2 J(x) + (\lambda + ce(x))^T \nabla^2 e(x) + c \nabla e(x)^T \nabla e(x) \in \mathbb{R}^{n \times n}.
\]
3.2 || The Advantage of the Expansion

For determining the minima of (ROP), we make use of

**Lemma 3.1.** Let $P, Q \in \mathbb{R}^{n \times n}$ be symmetric. Let $Q$ be positive semidefinite and $P$ be positive definite on the nullspace of $Q$, i.e. it holds $x^T P x > 0$ for all $x \in \mathbb{R}^n$ with $x \neq 0$ and $x^T Q x = 0$. Then there exists a $\bar{c} > 0$ such that the matrix $P + \bar{c} Q$ is positive definite for all $c > \bar{c}$.

**Proof.** A proof can be found in [Ber99, Chapter 3]. \hfill \Box

If $x^* \in \mathbb{R}^n$ and $\lambda^* \in \mathbb{R}^m$ satisfy the second-order sufficient optimality condition (see Theorem 2.7)

$$\nabla L(x^*, \lambda^*) = 0,$$

$$y^T \nabla^2_{xx} L(x^*, \lambda^*) y > 0 \text{ for all } y \neq 0 \text{ with } \nabla e(x^*) y = 0,$$

then it holds

$$\nabla^2 L_c(x^*, \lambda^*) = \nabla J(x^*) + (\lambda^* + c e(x^*))^T \nabla e(x^*)$$

$$= \nabla^2 L(x^*, \lambda^*) = 0$$

as well as

$$v^T \nabla^2_{xx} L_c(x^*, \lambda^*) v = v^T \left( \nabla^2 J(x^*) + (\lambda^* + c e(x^*))^T \nabla e(x^*) + c \nabla e(x^*)^T \nabla e(x^*) \right) v$$

$$= v^T \left( \nabla^2_{xx} L(x^*, \lambda^*) \right) v + c v^T \nabla e(x^*)^T \nabla e(x^*) v$$

$$\text{p.d. on nullspace of } \nabla e(x^*) = \|\nabla e(x^*) v\| \geq 0$$

for $v \in \mathbb{R}^n \setminus \{0\}$. According to Lemma 3.1, we get that $\nabla^2_{xx} L_c(x^*, \lambda^*)$ is positive definite for a $c > \bar{c}$ and simultaneously, $x^*$ is a strict local minimum of $L_c(x^*, \lambda^*)$.

**Example 3.2.** We compare the behaviour of the common Lagrangian method and the augmented Lagrangian method on the basis of the problem

$$\begin{align*}
\min J(x) &= -(x_1 x_2 + x_2 x_3 + x_1 x_3) \\
\text{s.t. } e(x) &= x_1 + x_2 + x_3 - 3 = 0.
\end{align*}$$

At first, we set up the Lagrange functions and calculate their gradients.

<table>
<thead>
<tr>
<th>Common Method</th>
<th>Augmented Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L(x, \lambda) = J(x) + \lambda e(x)$</td>
<td>$L_c(x, \lambda) = J(x) + \lambda e(x) + c e(x)^2$</td>
</tr>
</tbody>
</table>

$$\nabla L(x, \lambda)^T = \begin{pmatrix}
-x_2 - x_3 + \lambda \\
-x_1 - x_3 + \lambda \\
-x_1 - x_2 + \lambda \\
e(x)
\end{pmatrix}$$

$$\nabla L_c(x, \lambda)^T = \begin{pmatrix}
-x_2 - x_3 + \lambda + c e(x) \\
-x_1 - x_3 + \lambda + c e(x) \\
-x_1 - x_2 + \lambda + c e(x) \\
e(x)
\end{pmatrix}$$

By setting the gradients equal to zero, we get the same candidate for a minimum for both methods with the values

$$x_1^* = x_2^* = x_3^* = 1 \text{ and } \lambda^* = 2.$$
To check the sufficient condition of second order, we need the Hessian matrices.

<table>
<thead>
<tr>
<th>Common Method</th>
<th>Augmented Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nabla^2_{xx} L(x^<em>, \lambda^</em>) = \begin{pmatrix} 0 &amp; -1 &amp; -1 \ -1 &amp; 0 &amp; -1 \ -1 &amp; -1 &amp; 0 \end{pmatrix} )</td>
<td>( \nabla^2_{xx} L_c(x^<em>, \lambda^</em>) = \begin{pmatrix} c &amp; -1 + c &amp; -1 + c \ -1 + c &amp; c &amp; -1 + c \ -1 + c &amp; -1 + c &amp; c \end{pmatrix} )</td>
</tr>
</tbody>
</table>

On the left, for the common method, it remains to verify the positive definiteness on the nullspace of \( \nabla e(x^*) \). On the contrary, it can be easily proven that the matrix on the right is positive definite for all \( x \in \mathbb{R}^3 \setminus \{0\} \) and \( c > \bar{c} = \frac{2}{3} \).

### 3.3 Numerical Approach

In practice, instead of considering (ROP), we solve the *unrestricted* sequence of optimization problems

\[
(\text{OP}^k) \quad \min L_c(x, \lambda^k) \quad \text{s.t.} \quad x \in \mathbb{R}^n.
\]

By doing so, we obtain the wanted solution. This is evidenced by

**Lemma 3.3.** Let \( f \) and \( e \) be continuous and let \( \{ x \in \mathbb{R}^n \mid e(x) = 0 \} \) be non-empty. Let \( (x_k)_{k \in \mathbb{N}} \) be the sequence of the global minima of problem \( (\text{OP}^k) \), where \( (\lambda_k)_{k \in \mathbb{N}} \) is bounded, \( 0 < c^k < c^{k+1} \) holds for all \( k \in \mathbb{N} \) and \( c^k \to \infty \). Then, every limit point of the sequence \( (x_k)_{k \in \mathbb{N}} \) is a global minimum of the original problem (ROP).

**Proof.** A proof can be found in [Ber99, Chapter 4].

Our new problem \( (\text{OP}^k) \) has no constraints and it contains the penalty parameter \( c^k > 0 \). To find \( x^* \), we increase \( c^k \) in every iteration. A large \( c^k \) makes sense because it implies high costs of inadmissibility. As a result, \( x^* \) is admissible. For an admissible \( x \), we get

\[
L_c(x, \lambda) = J(x).
\]

Furthermore, it holds

\[
L_c(\tilde{x}, \lambda) \approx J(\tilde{x})
\]

for a nearly admissible \( \tilde{x} \). The \( x \) update is made by a gradient method (e.g. gradient descent, conjugate gradient method, etc.) to avoid bad condition. We set \( c^{k+1} = \beta c^k \) for \( \beta > 1 \) (\( \beta \in [4, 10] \) in practice (see [Ber99, page 405])) for the next iteration. We choose

\[
\| \nabla_x L_c(x^k, \lambda^k) \| \leq \epsilon^k \quad \text{for} \quad \epsilon^k \to 0
\]

as the stopping criterion. Last but not least, we update \( \lambda \) by means of

**Theorem 3.4.** Let \( J \) and \( e \) be continuously differentiable. Let

\[
0 < c^k < c^{k+1} \quad \text{and} \quad c^k \to \infty
\]

for \( (c^k)_{k \in \mathbb{N}} \) and let

\[
\epsilon^k \geq 0 \quad \text{and} \quad \epsilon^k \to 0
\]
for \((\epsilon^k)_{k \in \mathbb{N}}\). Let the sequence \((\lambda^k)_{k \in \mathbb{N}}\) be bounded and let \((x^k)_{k \in \mathbb{N}}\) satisfy
\[
\|\nabla_x L_c(x^k, \lambda^k)\| \leq \epsilon^k.
\]
Moreover, let the subsequence \((x^{k_j})_{j \in \mathbb{N}}\) converge to \(x^*\) such that \(\nabla e(x^*)\) has full rank. Then, \((\lambda^{k_j} + c^{k_j} e(x^{k_j}))_{j \in \mathbb{N}}\) converges to \(\lambda^*\) as well as
\[
\nabla J(x^*) + (\lambda^*)^T \nabla e(x^*) = 0 \quad \text{and} \quad e(x^*) = 0.
\]

**Proof.** A proof can be found in [Ber99, Chapter 4].

From that and because of the fact that the \(x\) update, which is done in the previous step, provides \(x^{k+1}\), the \(\lambda\) update is implemented by
\[
\lambda^{k+1} = \lambda^k + c^k e(x^{k+1}).
\]

The procedure discussed above can be implemented in the following way:

**Algorithm 1**: Augmented Lagrangian method

1. **Data:** Initial guess \(x^0\), initial multiplier \(\lambda^0\), weight \(c^0\), increment \(\beta\), tolerance \(\epsilon\);
2. **begin**
3. set \(k = 0\);
4. **while** \(\|\nabla L_c(x^k, \lambda^k)\| \geq \epsilon\) **do**
5. solve for \(x^{k+1}\) the problem (OP\(^k\));
6. update the Lagrange multiplier
\[
\lambda^{k+1} = \lambda^k + c^k e(x^{k+1});
\]
7. set \(c^{k+1} = \beta c^k\) and \(k = k + 1\);
8. **end.**

**Example 3.5.** Let us solve the problem
\[
\min J(x) = \frac{1}{2}(x_1^2 + x_2^2) \quad \text{s.t.} \quad e(x) = x_1 - 1 = 0.
\]

First, we calculate the solution by hand. The augmented Lagrange function reads
\[
L_c(x, \lambda) = \frac{1}{2}(x_1^2 + x_2^2) + \lambda(x_1 - 1) + \frac{c}{2}(x_1 - 1)^2.
\]

We obtain its gradient
\[
\nabla L_c(x, \lambda)^T = \begin{pmatrix} x_1 + \lambda + c(x_1 - 1) \\ x_2 \\ x_1 - 1 \end{pmatrix} = 0.
\]

This yields the candidate
\[
x^* = (1, 0) \quad \text{with} \quad \lambda^* = -1.
\]

Finally, the Hessian matrix is
\[
\nabla^2_{xx} L_c(x, \lambda) = \begin{pmatrix} 1 + c & 0 \\ 0 & 1 \end{pmatrix}.
\]
which is positive definite for all $c > 0$. Hence, $x^*$ is a minimum. Due to the contour lines we can guess the solution. In addition, one can see the effect of a constantly increasing $c$.

![Contour lines](image)

Figure 1: Contour lines (left: $L_{0.5}(\cdot,-3)$, right: $L_{20}(\cdot,-3)$)

The star in Figure 1 represents the known solution $x^*$. Now, we show that we get the same result for our numerical scheme. Therefore, we treat step 5 of Algorithm 1 with Algorithm 2:

**Algorithm 2**: Gradient descent method (for $f \in C^1$)

1: Data: Initial guess $x^0$;
2: begin
3: set $k = 0$;
4: while no convergence and $k < k_{max}$ do
5:    set descent direction $d^k = -\nabla f(x^k)$;
6:    choose step size $s_k \in (0, 1)$;
7:    calculate $x^{k+1} = x^k + s_k d^k$;
8:    set $k = k + 1$;
9: end.

We take $||x^{n+1} - x^n|| < \varepsilon_{gd}$ for a fixed $\varepsilon_{gd} > 0$ as the stopping criterion and choose the least integer $m$, such that the step size $s_k = \zeta^m$ for $\zeta \in (0, 1)$ fulfils the general sufficient decrease condition

$$f(x^{k+1}) - f(x^k) < -\alpha s_k ||\nabla f(x^k)||^2$$

(\alpha is typically set to $10^{-4}$) (see [Kel99, Chapter 3.2]), for the gradient descent method. Furthermore, by taking $\varepsilon = 10^{-5}, \varepsilon_{gd} = 0.2\varepsilon$ and $\zeta = 0.8$, we attain Table 1.

<table>
<thead>
<tr>
<th></th>
<th>$x_0$</th>
<th>$\lambda_0$</th>
<th>$c_0$</th>
<th>$\beta$</th>
<th>Iterations (Gradient loops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2, 1)</td>
<td>-2</td>
<td>1</td>
<td>10</td>
<td>${28, 80, 92, 6, 2}$</td>
</tr>
<tr>
<td>2</td>
<td>(2, 1)</td>
<td>-2</td>
<td>1</td>
<td>4</td>
<td>${28, 9, 18, 48, 12, 3}$</td>
</tr>
<tr>
<td>3</td>
<td>(2, 1)</td>
<td>-2</td>
<td>50</td>
<td>4</td>
<td>${112, 45, 6, 2}$</td>
</tr>
<tr>
<td>4</td>
<td>(2, 1)</td>
<td>-2</td>
<td>100</td>
<td>4</td>
<td>${1714, 22, 5, 2}$</td>
</tr>
<tr>
<td>5</td>
<td>$(-2 \cdot 10^5, -10^5)$</td>
<td>$2 \cdot 10^5$</td>
<td>1</td>
<td>4</td>
<td>${28, 9, 18, 48, 12, 3}$</td>
</tr>
<tr>
<td>6</td>
<td>$(-2 \cdot 10^5, -10^5)$</td>
<td>$2 \cdot 10^5$</td>
<td>100</td>
<td>10</td>
<td>${129, 8, 2}$</td>
</tr>
</tbody>
</table>

Table 1: Augmented Lagrangian method with different initial guesses
We note that the number of iterations in a single gradient loop (see Table 1, last column) tells us how long the gradient descent method needs to solve step 5 of Algorithm 1. The total number of gradient loops shows the number of how often the method repeats the while loop (step 4) until the stopping criterion is reached. For this example, every combination converges rounded to \( x^* = (1, 0), \lambda^* = -1 \) for all of the chosen initial guesses. The algorithm also converges for large \( x_0 \) and \( \lambda_0 \). A good coordination of the initial guesses \( c_0, \beta \) reduces the number of iterations in the single gradient loops (compare row 1 to row 4).

3.4 Problems with Inequality Constraints

The augmented Lagrangian method is also applicable for problems with inequality constraints. Let us consider the problem

\[
\min J(x) \quad \text{s.t.} \quad x \in \mathbb{R}^n, \quad e(x) = 0 \text{ and } g(x) \leq 0,
\]

where \( g : \mathbb{R}^n \to \mathbb{R}^p \) appears. First, using slack variables, we transform the inequality constraints to equality constraints and obtain the new problem

\[
\min J(x) \quad \text{s.t.} \quad x \in \mathbb{R}^n, \quad z \in \mathbb{R}^p, \quad e(x) = 0 \text{ and } g_i(x) + z_i^2 = 0,
\]

for \( i \in \{1, \ldots, p\} \). The corresponding augmented Lagrange function has the form

\[
\tilde{L}_c(x, z, \lambda, \mu) = J(x) + \lambda^T e(x) + \frac{c}{2} e(x)^T e(x) + \sum_{i=1}^{p} \mu_i (g_i(x) + z_i^2) + \frac{c}{2} \sum_{i=1}^{p} (g_i(x) + z_i^2)^2
\]

with the Lagrange multipliers \( \lambda \in \mathbb{R}^m \) and \( \mu \in \mathbb{R}^p \). Numerically, we apply the procedure above, i.e. we solve the problem

\[
\min_{x, z} \tilde{L}_c(x, z, \lambda, \mu) = J(x) + \lambda^T e(x) + \frac{c}{2} e(x)^T e(x) + \sum_{i=1}^{p} \mu_i (g_i(x) + z_i^2) + \frac{c}{2} \sum_{i=1}^{p} (g_i(x) + z_i^2)^2.
\]

At first, we minimize with respect to \( z \). Differentiating \( \tilde{L}_c(x, z, \lambda, \mu) \) and setting it to zero leads to the \( p \) equations

\[
\frac{\partial}{\partial z_i} \tilde{L}_c(x, z, \lambda, \mu) = 2 \mu_i z_i + c (g_i(x) + z_i^2) 2 z_i - \frac{1}{2} = 0 \quad \text{for } i = 1, \ldots, p.
\]

Dividing by \( 2z_i \) (in case \( z_i \neq 0 \)) and reordering yields

\[(z_i^*)^2 = - \left( \frac{\mu_i}{c} + g_i(x) \right) \quad \text{for } i = 1, \ldots, p.
\]

Adding the case \( z_i = 0 \), we get

\[(z_i^*)^2 = \max \left\{ 0, - \left( \frac{\mu_i}{c} + g_i(x) \right) \right\} \quad \text{for } i = 1, \ldots, p.
\]

Now, we can insert \((z_i^*)^2\) in \( \tilde{L}_c(x, z, \lambda, \mu) \) to obtain

\[
L_c(x, \lambda, \mu) := \tilde{L}_c(x, z^*, \lambda, \mu)
\]

\[
= J(x) + \lambda^T e(x) + \frac{c}{2} e(x)^T e(x) + \sum_{i=1}^{p} \left( \frac{\mu_i^2}{2c} \right) \chi_{\left\{ -\left( \frac{\mu_i}{c} + g_i(x) \right) > 0 \right\}}(x)
\]

\[
+ \sum_{i=1}^{p} \left( \mu_i g_i(x) + \frac{c}{2} g_i^2(x) \right) \chi_{\left\{ -\left( \frac{\mu_i}{c} + g_i(x) \right) \leq 0 \right\}}(x)
\]

12
where for a set $A$, it holds $\chi_A(x) = \begin{cases} 1 & \text{for } x \in A, \\ 0 & \text{otherwise} \end{cases}$. Finally, we minimize $L_c(x, \lambda, \mu)$ with respect to $x$ as is usual.
We consider the following state equation: For given
\[ u \in U_{ad} := \{ \tilde{u} \in L^2(0, T; \mathbb{R}^m) \mid u_a(t) \leq \tilde{u}(t) \leq u_b(t) \text{ in } [0, T] \text{ a.e.} \} \]
with \( u_a, u_b \in L^2(0, T; \mathbb{R}^m) \) the function \( y \) is governed by the linear parabolic differential equation with Robin boundary conditions
\begin{align*}
\text{(SEa)} & \quad y_t(t, x) - \lambda \Delta y(t, x) + v(t, x) \cdot \nabla y(t, x) = f(t, x) \quad \text{in } Q := (0, T) \times \Omega \text{ a.e.,} \\
\text{(SEb)} & \quad \lambda \frac{\partial y}{\partial \eta}(t, x) + \gamma_c y(t, x) = \gamma_c \sum_{i=1}^{m} u_i(t) b_i(x) \quad \text{in } \Sigma_c := (0, T) \times \Gamma_c \text{ a.e.,} \\
\text{(SEc)} & \quad \lambda \frac{\partial y}{\partial \eta}(t, x) + \gamma_{out} y(t, x) = \gamma_{out} y_{out}(t) \quad \text{in } \Sigma_{out} := (0, T) \times \Gamma_{out} \text{ a.e.,} \\
\text{(SEd)} & \quad y(0, x) = y_0(x) \quad \text{in } \Omega \text{ a.e.}
\end{align*}

The state equation (SE) is called convection-diffusion equation, heat-convection phenomena or heat equation with convection term. As its names indicate the equation models a combination of diffusion and convection. It characterizes physical phenomena where particles, energy or other physical quantities are conveyed inside a physical system due to the two aforementioned processes. In our case the convection-diffusion equation models the temperature distribution in a room \( \Omega \). There are \( m \) heaters in the room which are placed by the functions \( b_1, ..., b_m \). The respective usage of every heater is represented by the space-independent control functions \( u_1, ..., u_m \). Together, the right-hand side of (SEb) describes the heating process. The boundary \( \Gamma := \partial \Omega \) of the domain \( \Omega \) is split into the disjoint subsets \( \Gamma_c \) and \( \Gamma_{out} \) where it holds that
\[ \Gamma = \partial \Omega = \Gamma_c \cup \Gamma_{out}. \]
In particular, we assume that \( \Gamma_c \neq \emptyset \).

4.1 || Weak Formulation
At first, we introduce a weak formulation of (SE). For that, we multiply (SEa) by an arbitrary test function \( \varphi \in C^\infty(\Omega) \) and integrate over \( \Omega \) with the result
\[ \int_{\Omega} y_t(t, x) \varphi(x) \, dx - \lambda \int_{\Omega} \Delta y(t, x) \varphi(x) \, dx + \int_{\Omega} v(t, x) \cdot \nabla y(t, x) \varphi(x) \, dx = \int_{\Omega} f(t, x) \varphi(x) \, dx. \]
For the second term in (SEa), using Green’s first identity
\[ \int_{\Omega} \varphi \Delta \psi + \nabla \varphi \cdot \nabla \psi \, dx = \int_{\partial \Omega} \varphi \frac{\partial \psi}{\partial \eta} \, dA(x) \quad (\psi \in C^2(\Omega)) \]
in combination with inserting the boundary conditions \((SE_b), (SE_c)\), we get

\[
\lambda \int_{\Omega} \Delta y(t,x) \varphi(x) \, dx = -\lambda \int_{\Omega} \nabla y(t,x) \cdot \nabla \varphi(x) \, dx + \lambda \int_{\Gamma} \frac{\partial y}{\partial n}(t,x) \varphi(x) \, dA(x)
\]

\[
= -\lambda \int_{\Omega} \nabla y(t,x) \cdot \nabla \varphi(x) \, dx
\]

\[
+ \int_{\Gamma_c} \left( \gamma_c \sum_{i=1}^{m} u_i(t) b_i(x) - \gamma_c y(t,x) \right) \varphi(x) \, dA(x)
\]

\[
+ \int_{\Gamma_{out}} \left( \gamma_{out} y_{out}(t) - \gamma_{out} y(t,x) \right) \varphi(x) \, dA(x).
\]

This yields

\[
\int_{\Omega} y_t(t,x) \varphi(x) \, dx + \lambda \int_{\Omega} \nabla y(t,x) \cdot \nabla \varphi(x) \, dx + \gamma_c \int_{\Gamma_c} y(t,x) \varphi(x) \, dA(x)
\]

\[
+ \gamma_{out} \int_{\Gamma_{out}} y(t,x) \varphi(x) \, dA(x) + \int_{\Omega} v(t,x) \cdot \nabla y(t,x) \varphi(x) \, dx
\]

\[
= \int_{\Omega} f(t,x) \varphi(x) \, dx + \gamma_c \sum_{i=1}^{m} u_i(t) \int_{\Gamma_c} b_i(x) \varphi(x) \, dA(x) + \gamma_{out} y_{out}(t) \int_{\Gamma_{out}} \varphi(x) \, dA(x).
\]

Since \(C^\infty(\Omega) \subset H^1(\Omega)\) is dense, (4.1) also holds for all \(\varphi \in H^1(\Omega)\). Let \(V := H^1(\Omega)\) with \(V' = H^1(\Omega)^{'}\) and \(H := L^2(\Omega)\) be a Gelfand triple. By introducing the non-symmetric, time-dependent bilinear form \(a(t; \cdot, \cdot) : V \times V \to \mathbb{R}, \ t \in (0, T)\), and the mapping \(g : (0, T) \times V \to \mathbb{R}\) defined by

\[
a(t; \psi, \varphi) := \lambda \int_{\Omega} \nabla \psi(x) \cdot \nabla \varphi(x) \, dx + \gamma_c \int_{\Gamma_c} \psi(x) \varphi(x) \, dA(x) + \gamma_{out} \int_{\Gamma_{out}} \psi(x) \varphi(x) \, dA(x)
\]

\[
+ \int_{\Omega} v(t,x) \cdot \nabla \psi(x) \varphi(x) \, dx
\]

and

\[
g(t, \varphi) := \int_{\Omega} f(t,x) \varphi(x) \, dx + \gamma_c \sum_{i=1}^{m} u_i(t) \int_{\Gamma_c} b_i(x) \varphi(x) \, dA(x) + \gamma_{out} y_{out}(t) \int_{\Gamma_{out}} \varphi(x) \, dA(x),
\]

we obtain the weak formulation

\[
(SE_{w1a}) \quad y_t(t) + a(t; y(t), \cdot) = g(t, \cdot) \quad \text{in } V' \text{ a.e.,}
\]

\[
(SE_{w1b}) \quad y(0) = y_0 \quad \text{in } H.
\]

of \((SE)\) with the unknown

\[
y \in W(0,T) = \{ \varphi \in L^2(0, T; H^1(\Omega)) \mid \varphi_t \in L^2(0, T; H^1(\Omega)') \}. \]
4.2 || Well-Posedness and Unique Existence

Under the assumptions

- $\Omega \subset \mathbb{R}^n$ is a bounded domain with $C^1$-boundary $\Gamma$,
- diffusion parameter $\lambda > 0$,
- convection $v \in L^\infty(Q; \mathbb{R}^n)$ is time-dependent,
- $b = (b_1, \ldots, b_m)$ with $b_i \in L^2(\Gamma)$ for all $i \in \{1, \ldots, m\}$,
- isolation coefficients $\gamma_c, \gamma_{out} \geq 0$,
- $f \in L^2(Q)$,
- control $u$ is an element of the real Hilbert space $U \subset L^2(0, T; \mathbb{R}^m)$,
- the outer temperature $y_{out} \in L^2(0, T)$ is space-independent,
- initial temperature $y_0 \in L^2(\Omega)$,

we will show that (SE) has a unique solution using the following properties of the bilinear form $a$ and the right-hand side $g$.

**Lemma 4.1.** The mapping $\tilde{a} : V \to V', v \mapsto a(t; v, \cdot)$, $t \in (0, T)$, is well-defined and one has $\tilde{a} \in L(V, V')$.

**Proof.** Since $a$ is bilinear, the operator $\tilde{a}(\varphi)$ is linear for any $\varphi \in V$. For any $\varphi, \psi \in V$ and $t \in (0, T)$ using the Trace Theorem 2.8 and $\gamma_{max} := \max\{\gamma_c, \gamma_{out}\}$, we have

$$|a(t; \varphi, \psi)| = \lambda \int_\Omega \nabla \varphi(x) \nabla \psi(x) \, dx + \int_\Omega v(t, x) \cdot \nabla \varphi(x) \psi(x) \, dx + \gamma_c \int_{\Gamma_c} \varphi(x) \psi(x) \, dA(x)$$

$$+ \gamma_{out} \int_{\Gamma_{out}} \varphi(x) \psi(x) \, dA(x)$$

$$\leq \lambda \int_\Omega \nabla \varphi(x) \nabla \psi(x) \, dx + \int_\Omega v(t, x) \cdot \nabla \varphi(x) \psi(x) \, dx + \gamma_{max} \int_\Gamma \varphi(x) \psi(x) \, dA(x)$$

$$\leq \lambda \|\nabla \varphi\|_{L^2(\Omega; \mathbb{R}^n)} \|\nabla \psi\|_{L^2(\Omega; \mathbb{R}^n)} + \|v(t) \cdot \nabla \varphi\|_{L^2(\Omega)} \|\psi\|_{L^2(\Omega)}$$

$$+ \gamma_{max} \|\varphi\|_{L^2(\Gamma)} \|\psi\|_{L^2(\Gamma)}$$

$$\leq \lambda \|\varphi\|_V \|\psi\|_V + \|v\|_{L^\infty(\mathbb{R}^n)} \|\nabla \varphi\|_{L^2(\Omega; \mathbb{R}^n)} \|\psi\|_{L^2(\Omega)} + C_{TT}^2 \gamma_{max} \|\varphi\|_V \|\psi\|_V$$

$$\leq \lambda \|\varphi\|_V \|\psi\|_V + \|v\|_{L^\infty(\mathbb{R}^n)} \|\nabla \varphi\|_{L^2(\Omega; \mathbb{R}^n)} \|\psi\|_V + C_{TT}^2 \gamma_{max} \|\varphi\|_V \|\psi\|_V$$

$$= C \|\varphi\|_V \|\psi\|_V.$$

This means that $\tilde{a}$ is well-defined and $\tilde{a} \in L(V, V')$ with

$$\|\tilde{a}\|_{L(V, V')} \leq C := \lambda + \|v\|_{L^\infty(\mathbb{R}^n)} + C_{TT}^2 \gamma_{max}.$$
Lemma 4.2. The bilinear form $a$ is coercive with coercivity constants $\alpha = \frac{\lambda}{2}$ and $\beta = \frac{\|\cdot\|_\infty^2}{2\lambda} + \frac{\lambda}{2}$.

Proof. We take an arbitrary $\varphi \in V$ and define $C_v := \|v\|_{L^\infty(Q;\mathbb{R}^n)}$. Since it holds $\gamma_c, \gamma_{\text{out}} \geq 0$ and by using Young’s inequality $(ab \leq \frac{a^2}{2\varepsilon} + \frac{b^2}{2}$ for $a, b \in \mathbb{R}_{\geq 0}, \varepsilon > 0)$, we get

$$a(t; \varphi, \varphi) \geq \lambda \int_\Omega \nabla \varphi(x) \nabla \varphi(x) \, dx + \int_\Omega (v(t, x) \cdot \nabla \varphi(x)) \varphi(x) \, dx$$

for $t \in (0, T)$.

Lemma 4.3. The right-hand side $\tilde{g} : (0, T) \rightarrow V'$, $t \mapsto g(t, \cdot)$ is well-defined and it holds $\tilde{g} \in L^2(0, T; V')$.

Proof. When we look at the definition of $g$, it is obvious that $\tilde{g}(t) \in V'$ is a linear operator for all $t \in (0, T)$. Applying the Trace Theorem 2.8, for almost any $t \in (0, T)$ and any $\varphi \in V$, it holds

$$|\tilde{g}(t) \varphi| = \left| \int f(t, x) \varphi(x) \, dx + \gamma_c \sum_{i=1}^m u_i(t) \int b_i(x) \varphi(x) \, dA(x) + \gamma_{\text{out}} y_{\text{out}}(t) \int \varphi \, dA(x) \right|$$

$$\leq \|f\|_{L^2(\Omega)} \|\varphi\|_{L^2(\Omega)} + \gamma_c \sum_{i=1}^m |u_i(t)| \int |b_i(x)\varphi(x)| \, dA(x) + \gamma_{\text{out}} |y_{\text{out}}(t)| \int |\varphi| \, dA(x)$$

$$\leq \|f\|_{L^2(\Omega)} \|\varphi\|_{L^2(\Omega)} + \gamma_c \sum_{i=1}^m |u_i(t)| \|b_i\|_{L^2(\Gamma)} \|\varphi\|_{L^2(\Gamma)} \sqrt{\text{Leb}(\Gamma)}$$

$$+ \gamma_{\text{out}} |y_{\text{out}}(t)| \|\varphi\|_{L^2(\Gamma)} \sqrt{\text{Leb}(\Gamma)}$$

$$\leq \|f\|_{L^2(\Omega)} \|\varphi\|_{V} + \gamma_C_{TT} \sqrt{\text{Leb}(\Gamma)} \sum_{i=1}^m |u_i(t)| \|b_i\|_{L^2(\Gamma)} \|\varphi\|_{V}$$

$$+ \gamma_{\text{out}} C_{TT} \sqrt{\text{Leb}(\Gamma)} |y_{\text{out}}(t)| \|\varphi\|_{V}$$

$$= : C(t) \|\varphi\|_{V}$$

where $\text{Leb}(\Gamma)$ is the Lebesgue measure of $\Gamma$. Furthermore, we get $\|\tilde{g}\|_{V'} \leq C(t)$ and

$$\|\tilde{g}\|_{L^2(0, T; V')} = \int_0^T \|\tilde{g}(t)\|_{V'}^2 \, dt \leq \int_0^T C(t)^2 \, dt$$
\[
\begin{align*}
&= \int_0^T \left( \| f \|_{L^2(\Omega)} + \gamma_c C_{TT} \sqrt{\text{Leb}(\Gamma)} \sum_{i=1}^m |u_i(t)| \| b_i \|_{L^2(\Gamma)} \\
&\quad + \gamma_{out} C_{TT} \sqrt{\text{Leb}(\Gamma)} |y_{out}(t)| \right)^2 \, dt \\
&\leq \int_0^T 2 \left( \| f \|_{L^2(\Omega)} + \gamma_c C_{TT} \sqrt{\text{Leb}(\Gamma)} \sum_{i=1}^m |u_i(t)| \| b_i \|_{L^2(\Gamma)} \right)^2 \, dt \\
&\quad + \int_0^T 2 \left( \gamma_{out} C_{TT} \sqrt{\text{Leb}(\Gamma)} |y_{out}(t)| \right)^2 \, dt \\
&\leq 4T \| f \|_{L^2(\Omega)}^2 + 4 \int_0^T \left( \gamma_c C_{TT} \sqrt{\text{Leb}(\Gamma)} \sum_{i=1}^m \| b_i \|_{L^2(\Gamma)} |u_i(t)| \right)^2 \, dt \\
&\quad + 2\gamma_{out}^2 C_{TT}^2 \sqrt{\text{Leb}(\Gamma)} \int_0^T |y_{out}(t)|^2 \, dt \\
&\leq 4T \| f \|_{L^2(\Omega)}^2 + 4\gamma_c^2 C_{TT}^2 \sqrt{\text{Leb}(\Gamma)} \| b \|_{L^2(\Gamma;\mathbb{R}^m)}^2 \int_0^T \| u(t) \|_{\mathbb{R}^m}^2 \, dt \\
&\quad + 2\gamma_{out}^2 C_{TT}^2 \sqrt{\text{Leb}(\Gamma)} \| y_{out} \|_{L^2(\Omega;\mathbb{R}^m)}^2 \\
&= 4T \| f \|_{L^2(\Omega)}^2 + 4\gamma_c^2 C_{TT}^2 \sqrt{\text{Leb}(\Gamma)} \| b \|_{L^2(\Gamma;\mathbb{R}^m)}^2 \| u \|_{U}^2 + 2\gamma_{out}^2 C_{TT}^2 \sqrt{\text{Leb}(\Gamma)} \| y_{out} \|_{L^2(\Omega;\mathbb{R}^m)}^2 \\
&< \infty,
\end{align*}
\]
where we used \((r + s)^2 \leq 2r^2 + 2s^2\) \((r, s \in \mathbb{R}_{\geq 0})\) twice. This shows the well-definedness of \(\tilde{g}\) and \(\hat{g} \in L^2(0, T; V')\).

**Theorem 4.4.** Under the given assumptions \((SE)\) has a unique weak solution \(y \in W(0, T)\) satisfying

\[\| y \|_{W(0, T)} \leq C \left( \| y_0 \|_H + \| \tilde{g} \|_{L^2(0, T; V')} \right).\]

**Proof.** This follows directly from Theorem 2.10 since \(V = H^1(\Omega)\) and \(H = L^2(\Omega)\) form a Gelfand triple, the bilinear form \(a\) is coercive and it holds \(\hat{g} \in L^2(0, T; V')\) and \(y_0 \in L^2(\Omega)\).

This leads us to

**Definition and Remark 4.5.** We define the solution operator

\[\mathcal{T} : U \to W(0, T)\]

with \(y = \mathcal{T} u\) as the weak solution to \((SE)\). Moreover, we split the solution

\[(\mathcal{T} u)(t, x) = y(t, x) := \hat{y}(t, x) + (\mathcal{S} u)(t, x)\]

in the inhomogeneous part \(\hat{y}\) and the homogeneous part \(\mathcal{S} u\) where \(\hat{y}\) solves \((SE)\) for \(u = 0\) and \(\mathcal{S} u\) solves \((SE)\) for \(f = 0, \gamma_{out} y_{out} = 0\) and \(y_0 = 0\) in the weak sense, respectively. We notice that \(\mathcal{S}\) is well-defined, linear and bounded. Hence, the adjoint \(\mathcal{S}^* : W(0, T) \to U\) exists.
Chapter 5

THE OPTIMAL CONTROL PROBLEM OF THE CONVECTION-DIFFUSION MODEL

The goal is to track the temperature \( y \) to given desired temperature profiles \( y_Q \in L^2(Q) \) and \( y_T \in H \) while utilizing as less control input as possible. Therefore, we introduce the cost function \( J : W(0, T) \times U \to \mathbb{R} \) with

\[
J(y, u) = \frac{\sigma_Q}{2} \int_0^T \| y(t) - y_Q(t) \|_H^2 \, dt + \frac{\sigma_T}{2} \| y(T) - y_T \|_H^2 + \frac{1}{2} \sum_{i=1}^m \sigma_i \| u_i \|_{L^2(0, T)}^2,
\]

where \( \sigma_Q, \sigma_T \geq 0 \) and \( \sigma_i > 0, i = 1, \ldots, m \), are weighting parameters. We define the reduced cost functional by

\[
\hat{J}(u) = J(Tu, u) \quad \text{for } u \in U_{ad}.
\]

This makes sense since the room temperature \( y \) is controlled by \( u \) in the way that \( u \) specifies how much the respective heater is heating in each time step. One can also say that the control input \( u \) represents the heating and the controlled outcome is the temperature \( y \) in the room.

In order to achieve our above-said goal, we consider the optimal boundary control problem with state constraints

\[
\text{(OCP)} \quad \min \hat{J}(u) \quad \text{s.t. } u \in U_{ad}, \quad y_a \leq Tu \leq y_b \text{ in } Q \text{ a.e.},
\]

where \( y_a, y_b \in L^2(Q) \) are given lower and upper bounds, respectively. We define the slack variables \( s_a := Tu - y_a \) and \( s_b := y_b - Tu \). As a consequence, we have

\[
\begin{align*}
(5.2) \quad y_a - Tu + s_a &= 0 \quad \text{in } Q \text{ a.e.}, \\
(5.3) \quad Tu - y_b + s_b &= 0 \quad \text{in } Q \text{ a.e.},
\end{align*}
\]

together with \( s_a \geq 0 \) and \( s_b \geq 0 \). By doing so, (OCP) turns into an unrestricted optimization problem like the ones we faced in Chapter 3. For this kind of problem, we introduced the augmented Lagrangian method. In the following, we set up the augmented Lagrange function and compute its derivative. In the upcoming algorithm, we minimize the augmented Lagrange function with a first-order method. This means that we only require the already determined first derivative. To deal with the gradient, we introduce the adjoint equation. This adjoint equation is a partial differential equation which comprises some parts from the state equation and whose solution is one part of the gradient - namely, the part which is numerically not practicable without providing the adjoint equation.
5.1 || The Augmented Lagrange Function of the Optimal Control Problem

According to (3.1), for $c > 0$, the augmented Lagrange function is given as

$$\mathcal{L}_c(u, s_a, s_b, \mu_a, \mu_b) = \hat{J}(u) + \langle \mu_a, y_a - \mathcal{T}u + s_a \rangle_{L^2(Q)} + \langle \mu_b, \mathcal{T}u - y_b + s_b \rangle_{L^2(Q)}$$

$$+ \frac{c}{2} \|y_a - \mathcal{T}u + s_a\|^2_{L^2(Q)} + \frac{c}{2} \|\mathcal{T}u - y_b + s_b\|^2_{L^2(Q)}.$$ 

considering (5.2), (5.3) as equality constraints. From the optimality conditions

$$\frac{\partial \mathcal{L}_c}{\partial s_a}(u, s_a, s_b, \mu_a, \mu_b) \delta_{s_a} = \langle \mu_a, \delta_{s_a} \rangle_{L^2(Q)} + c \langle y_a - \mathcal{T}u + s_a, \delta_{s_a} \rangle_{L^2(Q)} = 0$$

and

$$\frac{\partial \mathcal{L}_c}{\partial s_b}(u, s_a, s_b, \mu_a, \mu_b) \delta_{s_b} = \langle \mu_b, \delta_{s_b} \rangle_{L^2(Q)} + c \langle \mathcal{T}u - y_b + s_b, \delta_{s_b} \rangle_{L^2(Q)} = 0$$

for $\delta_{s_a}, \delta_{s_b} \in L^2(Q)$, we derive the two equalities

$$\mu_a + c(y_a - \mathcal{T}u + s_a) = 0 \quad \text{in } Q \text{ a.e.},$$

$$\mu_b + c(\mathcal{T}u - y_b + s_b) = 0 \quad \text{in } Q \text{ a.e.}$$

To ensure $s_a \geq 0$ and $s_b \geq 0$ we set

(5.4) \hspace{1cm} s_a = \max \left\{ 0, \mathcal{T}u - y_a - \frac{1}{c}\mu_a \right\},

(5.5) \hspace{1cm} s_b = \max \left\{ 0, y_b - \mathcal{T}u - \frac{1}{c}\mu_b \right\}.

Let $Q^+_a, Q^+_b, Q^-_a, Q^-_b$ be defined as

$$Q^+_a = \left\{ (t, x) \left| \left( \mathcal{T}u - y_a - \frac{1}{c}\mu_a \right)(t, x) > 0 \text{ a.e.} \right. \right\},$$

$$Q^+_b = \left\{ (t, x) \left| \left( y_b - \mathcal{T}u - \frac{1}{c}\mu_b \right)(t, x) > 0 \text{ a.e.} \right. \right\},$$

$$Q^-_a = Q \setminus Q^+_a, \quad Q^-_b = Q \setminus Q^+_b.$$ 

Then, we find

$$\mathcal{L}_c(u, s_a, s_b, \mu_a, \mu_b) = \hat{J}(u) + \int_{Q^+_a} \mu_a \left( y_a - \mathcal{T}u + \left( \mathcal{T}u - y_a - \frac{1}{c}\mu_a \right) \right) \, dV(t, x)$$

$$+ \frac{c}{2} \int_{Q^+_a} \left( y_a - \mathcal{T}u + \left( \mathcal{T}u - y_a - \frac{1}{c}\mu_a \right) \right)^2 \, dV(t, x)$$

$$+ \int_{Q^-_a} \mu_a(y_a - \mathcal{T}u - 0) \, dV(t, x) + \frac{c}{2} \int_{Q^-_a} (y_a - \mathcal{T}u - 0)^2 \, dV(t, x)$$

$$+ \int_{Q^+_b} \mu_b \left( \mathcal{T}u - y_b + \left( y_b - \mathcal{T}u - \frac{1}{c}\mu_b \right) \right) \, dV(t, x)$$

$$+ \frac{c}{2} \int_{Q^+_b} \left( \mathcal{T}u - y_b + \left( y_b - \mathcal{T}u - \frac{1}{c}\mu_b \right) \right)^2 \, dV(t, x)$$

$$+ \int_{Q^-_b} \mu_b(\mathcal{T}u - y_b - 0) \, dV(t, x) + \frac{c}{2} \int_{Q^-_b} (\mathcal{T}u - y_b - 0)^2 \, dV(t, x)$$

20
\[ + \frac{c}{2} \int_{Q_a^+} \left( \mathcal{T} u - y_b + \left( y_b - \mathcal{T} u - \frac{1}{c} \mu_b \right) \right)^2 dV(t, x) \]
\[ + \int_{Q_a^-} \mu_a (\mathcal{T} u - y_a - 0) \, dV(t, x) + \frac{c}{2} \int_{Q_b^-} (\mathcal{T} u - y_b - 0)^2 \, dV(t, x) \]
\[ = \hat{J}(u) + \int_{Q_a^+} -\frac{1}{c} \mu_a^2 + \frac{c}{2} \left( \frac{1}{c} \mu_a \right)^2 \, dV(t, x) \]
\[ + \int_{Q_a^-} \mu_a (y_a - \mathcal{T} u) + \frac{c}{2} (y_a - \mathcal{T} u)^2 \, dV(t, x) \]
\[ + \int_{Q_b^+} \mu_b \left( -\frac{1}{c} \mu_b \right) + \frac{c}{2} \left( \frac{1}{c} \mu_b \right)^2 \, dV(t, x) \]
\[ + \int_{Q_b^-} \mu_b (\mathcal{T} u - y_b) + \frac{c}{2} (\mathcal{T} u - y_b)^2 \, dV(t, x) \]
\[ = \hat{J}(u) - \frac{1}{2c} \int_Q \mu_a^2 \, dV(t, x) - \frac{1}{2c} \int_Q \mu_b^2 \, dV(t, x) \]
\[ + \int_{Q_a^+} \frac{1}{2c} \mu_a^2 + \mu_a (y_a - \mathcal{T} u) + \frac{c}{2} (y_a - \mathcal{T} u)^2 \, dV(t, x) \]
\[ + \int_{Q_a^-} \mu_b \left( -\frac{1}{c} \mu_b \right) + \frac{c}{2} \left( \frac{1}{c} \mu_b \right)^2 \, dV(t, x) \]
\[ + \int_{Q_b^+} \mu_b (\mathcal{T} u - y_b) + \frac{c}{2} (\mathcal{T} u - y_b)^2 \, dV(t, x) \]
\[ = \hat{J}(u) - \frac{1}{2c} \||\mu_a||^2_{L^2(Q)} - \frac{1}{2c} \||\mu_b||^2_{L^2(Q)} \]
\[ + \frac{c}{2} \int_{Q_a^+} \left( \frac{\mu_a}{c} \right)^2 + 2 \left( \frac{\mu_a}{c} \right) (y_a - \mathcal{T} u) + (y_a - \mathcal{T} u)^2 \, dV(t, x) \]
\[ + \frac{c}{2} \int_{Q_b^+} \left( \frac{\mu_b}{c} \right)^2 + 2 \left( \frac{\mu_b}{c} \right) (\mathcal{T} u - y_b) + (\mathcal{T} u - y_b)^2 \, dV(t, x) \]
\[ = \hat{J}(u) - \frac{1}{2c} \||\mu_a||^2_{L^2(Q)} - \frac{1}{2c} \||\mu_b||^2_{L^2(Q)} \]
\[ + \frac{c}{2} \int_{Q_a^-} \left( y_a - \mathcal{T} u + \frac{\mu_a}{c} \right)^2 \, dV(t, x) \]
\[ + \frac{c}{2} \int_{Q_b^-} \left( \mathcal{T} u - y_b + \frac{\mu_b}{c} \right)^2 \, dV(t, x) \]

Now, we use that \( y_a - \mathcal{T} u + \frac{\mu_a}{c} < 0 \) on \( Q_a^+ \) and \( \mathcal{T} u - y_b + \frac{\mu_b}{c} < 0 \) on \( Q_b^+ \). Hence, it holds
\[
L_c(u, s_a, s_b, \mu_a, \mu_b) = \hat{J}(u) - \frac{1}{2c} ||\mu_a||^2_{L^2(Q)} - \frac{1}{2c} ||\mu_b||^2_{L^2(Q)}
\]
By having this, we can formulate

**Algorithm 3**: First-order augmented Lagrangian method

1. **Data**: Initial pair \((\mu_a^0, \mu_b^0) \in L^2(Q) \times L^2(Q)\), initial weight \(c_0 > 0\), increment \(\beta > 0\) for \(c_n\), tolerance \(\varepsilon > 0\), \(n_{\text{max}}\) maximum number of iterations;
2. **begin**
3. set \(n = 0\) and \(\text{FLAG} = \text{true}\);
4. **while** \(\text{FLAG}\) and \(n < n_{\text{max}}\) **do**
5. for fixed \((\mu_a^n, \mu_b^n)\) find \(u^{n+1}\) solving the problem

\[
\min_{u} L_{c_n}(u, \mu_a^n, \mu_b^n) = \hat{J}(u) + \frac{c_n}{2} \left\| \max \left\{ 0, y_a - T u + \frac{\mu_a^n}{c_n} \right\} \right\|_{L^2(Q)}^2
\]

\[
+ \frac{1}{2c_n} \left( \|\mu_a^n\|_{L^2(Q)}^2 + \|\mu_b^n\|_{L^2(Q)}^2 \right)
\]

\[
\quad \text{s.t. } u \in U_{\text{ad}};
\]

6. update the Lagrange multipliers

\[
\mu_a^{n+1} = \max \{ 0, \mu_a^n + c_n(0 - Tu^{n+1}) \},
\]

\[
\mu_b^{n+1} = \max \{ 0, \mu_b^n + c_n(T u^{n+1} - y_b) \};
\]

7. **if** \(\|u^n - u^{n+1}\|_U \leq \varepsilon\) **then**
8. \(\text{FLAG} = \text{false};\)
9. set \(c_{n+1} = \beta c_n, n = n + 1;\)
10. **end.**

Let us verify the multipliers’ update. After introducing the slack variables, we have the equality constraints (5.2) and (5.3). We insert (5.4) into (5.2) and (5.5) into (5.3),
respectively, and obtain

\[ e_a(u) := y_a - \mathcal{T} u + \max \left\{ 0, \mathcal{T} u - y_a - \frac{1}{c} \mu_a \right\} = 0 \]

\[ e_b(u) := \mathcal{T} u - y_b + \max \left\{ 0, y_b - \mathcal{T} u - \frac{1}{c} \mu_b \right\} = 0 \]

which in turn allows us to write

\[ \mu_a^{n+1} = \mu_a^n + c_n e_a(u^{n+1}) \]

\[ = \mu_a^n + c_n \left( y_a - \mathcal{T} u^{n+1} + \max \left\{ 0, \mathcal{T} u^{n+1} - y_a - \frac{1}{c} \mu_a^n \right\} \right) \]

\[ = \begin{cases} 
\mu_a^n + c_n (y_a - \mathcal{T} u^{n+1}) & \text{if } \max \left\{ 0, \mathcal{T} u^{n+1} - y_a - \frac{1}{c} \mu_a^n \right\} = 0, \\
0 & \text{otherwise} 
\end{cases} \]

\[ = \max \{0, \mu_a^n + c_n (y_a - \mathcal{T} u^{n+1})\} \]

and

\[ \mu_b^{n+1} = \max \{0, \mu_b^n + c_n (\mathcal{T} u^{n+1} - y_b)\}, \]

analogously.

### 5.2 The Derivative of the Augmented Lagrange Function

Now, let us compute the derivatives of

\[ \hat{L}_c(u, \mu_a, \mu_b) = \hat{J}(u) + \frac{c}{2} \left( \left\| \max \left\{ 0, y_a - \mathcal{T} u + \frac{\mu_a}{c} \right\} \right\|_{L^2(Q)}^2 + \left\| \max \left\{ 0, \mathcal{T} u - y_b + \frac{\mu_b}{c} \right\} \right\|_{L^2(Q)}^2 \right) - \frac{1}{2c} (\| \mu_a \|_{L^2(Q)}^2 + \| \mu_b \|_{L^2(Q)}^2). \]

Differentiation with respect to \( u \) yields

\[ \frac{\partial \hat{L}_c}{\partial u}(u, \mu_a, \mu_b) u^\delta = \hat{J}'(u) u^\delta + c \left( \max \left\{ 0, y_a - \mathcal{T} u + \frac{\mu_a}{c} \right\}, -\mathcal{T}'(u) u^\delta \right)_{L^2(0,T;H)} \]

\[ + c \left( \max \left\{ 0, \mathcal{T} u - y_b + \frac{\mu_b}{c} \right\}, \mathcal{T}'(u) u^\delta \right)_{L^2(0,T;H)} \]

\[ = \left\langle \nabla \hat{J}(u) + c \mathcal{T}'(u)^* \left( \max \left\{ 0, \mathcal{T} u - y_b + \frac{\mu_b}{c} \right\} \right) -\max \left\{ 0, y_a - \mathcal{T} u + \frac{\mu_a}{c} \right\}, u^\delta \right\rangle_U \]

for \( u^\delta \in U \). Let \( e_i \) \((i = 1, ..., m)\) be the standard basis for the Euclidean vector space \( \mathbb{R}^m \).

Looking more precisely at the first summand

\[ \left\langle \nabla \hat{J}(u), u^\delta \right\rangle_U = \left\langle \nabla_y J(y, u), \mathcal{T}'(u) u^\delta \right\rangle_{L^2(0,T;H)} + \left\langle \nabla_u J(y, u), u^\delta \right\rangle_U \]

23
\[
\n\begin{align*}
\n\frac{\partial \hat{L}_c}{\partial u}(u, \mu_a, \mu_b) & = \left\langle \sigma_Q (y - y_Q) + c \left( \max \left\{ 0, T u - y_b + \frac{\mu_b}{c} \right\} \right), \mathcal{T}'(u) u^\delta \right\rangle_{L^2(0,T;H)} \\
& \quad - \left\langle \sigma_Q (y - y_Q) - c \left( \max \left\{ 0, y_a - T u + \frac{\mu_a}{c} \right\} \right), \mathcal{T}'(u) u^\delta \right\rangle_{L^2(0,T;H)} \\
& \quad + \left\langle \sigma_T (y(T) - y_T), \mathcal{T}'(u) u^\delta(T) \right\rangle_H + \left\langle \left( \begin{array}{c} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{array} \right), u^\delta \right\rangle_U \\
& = \left\langle \sigma_Q (y - y_Q) + c \left( \max \left\{ 0, T u - y_b + \frac{\mu_b}{c} \right\} \right), \mathcal{T}'(u) u^\delta \right\rangle_{L^2(0,T;H)} \\
& \quad - \left\langle \sigma_Q (y - y_Q) - c \left( \max \left\{ 0, y_a - T u + \frac{\mu_a}{c} \right\} \right), \mathcal{T}'(u) u^\delta \right\rangle_{L^2(0,T;H)} \\
& \quad + \left\langle \sigma_T (y(T) - y_T), \mathcal{T}'(u) u^\delta(T) \right\rangle_H + \left\langle \left( \begin{array}{c} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{array} \right), u^\delta \right\rangle_U \\
& \quad + \left\langle \sigma_T (y(T) - y_T), \mathcal{S} u^\delta(T) \right\rangle_H + \left\langle \left( \begin{array}{c} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{array} \right), u^\delta \right\rangle_U.
\end{align*}
\]

Using the adjoint operator \( \mathcal{S}^* : W(0,T) \to U \) of the solution operator \( \mathcal{S} : U \to W(0,T) \), we obtain
\[
\frac{\partial \hat{L}_c}{\partial u}(u, \mu_a, \mu_b)u^\delta = \left\langle S^\star \left( \sigma_Q (y - y_Q) + c \max \left\{ 0, T u - y_b + \mu_b \right\} \right), u^\delta \right\rangle_U \\
- c \max \left\{ 0, y_a - T u + \mu_a \right\} + \left( \begin{array}{c} \sigma_{1u_1} \\ \sigma_{2u_2} \\ \vdots \\ \sigma_{mu_m} \end{array} \right), u^\delta \right\rangle_U \\
+ \left\langle \sigma_T (y(T) - y_T), S u^\delta(T) \right\rangle_H
\]

(5.8)

for \( u^\delta \in U \). In the previous computation we made use of the Hilbert adjoint \( S^\star : W(0,T) \rightarrow U \) of the homogeneous solution operator \( S \). Since we do not know how the adjoint \( S^\star \) looks like and how to evaluate \( S^\star \), we introduce the adjoint equation in the next chapter.

5.3 \| The Adjoint Equation

For given \( u \), we solve the state equation (SE) to get \( y = T u \) and define

\[
Q_a = \left\{ (t, x) \in Q \left| \left( y_a - y + \frac{\mu_a}{c} \right)(t, x) > 0 \right. \right\}, \\
Q_b = \left\{ (t, x) \in Q \left| \left( y - y_b + \frac{\mu_b}{c} \right)(t, x) > 0 \right. \right\}. 
\]

For computing parts of \( \frac{\partial \hat{L}_c}{\partial a} \), we introduce a second partial differential equation in

**Definition 5.1.** For the weak formulation (SE\_wf) and a given \( u \in U_{ad} \), the end value problem

(\( AE \))

\[-p_i(t) + a(t; \cdot, p(t)) = \sigma_Q (y_Q(t) - T u(t)) + c \left( y_a(t) - T u(t) + \frac{\mu_a(t)}{c} \right) \chi_{Q_a} \\
- c \left( T u(t) - y_b(t) + \frac{\mu_b(t)}{c} \right) \chi_{Q_b} \quad \text{in} \ V' \ a.e., \\
p(T) = \sigma_T (y_T - T u(T)) \quad \text{in} \ H,
\]

is called the *adjoint equation*.

The reason why the right-hand side of the adjoint has this form is the derivative \( \frac{\partial \hat{L}_c}{\partial a} \) of the augmented Lagrange function: We want to get rid of the first term - the one which includes the adjoint \( S^\star \) in \( \frac{\partial \hat{L}_c}{\partial a} \). In the following we will see how the adjoint equation helps to achieve that. We point out that the adjoint equation has an additional right-hand side compared to the one in [MV18, pages 7-8].

**Lemma 5.2.** The adjoint equation (AE) admits a unique solution \( p \in W(0,T) \) for each \( u \in U_{ad} \). In particular, there exists a \( C > 0 \) such that

\[
\| p \|_{W(0,T)} \leq C \left( \| \sigma_Q (y_Q - T u) + c \left( y_a - T u + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( T u - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \|_{L^2(0,T;V')} \\
+ \| \sigma_T (y_T - T u(T)) \|_{H} \right).
\]
Proof. We define \( \tilde{p}(t) := p(T - t) \) for all \( t \in [0, T] \) to transform \( \text{(AE)} \) into the initial value problem

\[
\begin{align*}
-\tilde{p}_t(t) + a(t; \cdot, \tilde{p}(t)) &= \sigma_Q(y_Q(T - t) - \mathcal{T} u(T - t)) \\
&\quad + c \left( y_a(T - t) - \mathcal{T} u(T - t) + \frac{\mu_a(t)}{c} \right) \chi_{Q_a} \\
&\quad - c \left( \mathcal{T} u(T - t) - y_b(T - t) + \frac{\mu_b(t)}{c} \right) \chi_{Q_b} \quad \text{in } V' \text{ a.e.,}
\end{align*}
\]

(5.9)

Lemma 4.2 tells us that \( a \) is coercive. Since

\[
\sigma_Q(y_Q(T - \cdot) - \mathcal{T} u(T - \cdot)) + c \left( y_a(T - \cdot) - \mathcal{T} u(T - \cdot) + \frac{\mu_a(\cdot)}{c} \right) \chi_{Q_a} \\
- c \left( \mathcal{T} u(T - \cdot) - y_b(T - \cdot) + \frac{\mu_b(\cdot)}{c} \right) \chi_{Q_b} \in L^2(0, T; V)
\]

and \( \sigma_T(y_T - \mathcal{T} u(T)) \in H \), we can apply Theorem 2.10. Consequently, there exists a unique solution \( \tilde{p} \) to (5.9) for each \( u \in U_{ad} \) with

\[
\|\tilde{p}\|_{W(0, T)} \leq C \left( \|\sigma_Q(y_Q - \mathcal{T} u) + c \left( y_a - \mathcal{T} u + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( \mathcal{T} u - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \|_{L^2(0, T; V')} \\
+ \|\sigma_T(y_T - \mathcal{T} u(T))\|_H \right).
\]

But then, there is a unique solution \( p \) of \( \text{(AE)} \) which fulfills the same inequality. \( \quad \square \)

Definition and Remark 5.3. We split the right-hand side \( g(t, \varphi) := r(t)\varphi + (\mathcal{B} u)(t)\varphi \) of \( \text{(SEwt)} \) analogously to Definition and Remark 4.5 in the inhomogeneous part \( r : (0, T) \to V' \),

\[
r(t)\varphi := \int_{\Omega} f(t, x)\varphi(x) \, dx + \gamma_{out} \int_{\Gamma_{out}} \varphi(x) \, dA(x) \quad (t \in (0, T), \varphi \in V)
\]

and the homogeneous part \( \mathcal{B} : U \to L^2(0, T; V') \),

\[
(\mathcal{B} u)(t)\varphi := \gamma_c \sum_{i=1}^m u_i(t) \int_{\Gamma_c} b_i(x)\varphi(x) \, dA(x) \quad (t \in (0, T), u \in U, \varphi \in V).
\]

Considering the proof of Lemma 4.3, we can similarly compute that there exists a \( C > 0 \) such that

\[
\|\mathcal{B} u\|_{L^2(0, T; V')} \leq C \|b\|_{L^2(\Gamma; \mathbb{R}^m)} \|u\|_U.
\]

Definition 5.4. We define the linear and continuous solution operator \( \mathcal{R}_1 : L^2(0, T; V') \to W(0, T) \) which maps an arbitrary right-hand side \( f \in L^2(0, T; V') \) of

\[
\begin{align*}
-p_t(t) + a(t; \cdot, p(t)) &= f(t) \quad \text{in } V' \text{ a.e.,} \\
p(T) &= \sigma_1(y_T - \hat{y}(T)) \quad \text{in } H,
\end{align*}
\]

(5.10)
to its unique solution. Furthermore, we define another linear and continuous solution operator $R_2 : L^2(0,T;V') \to W(0,T)$ which maps an arbitrary right-hand side $f \in L^2(0,T;V')$ of

\begin{equation}
- p_t(t) + a(t; \cdot, p(t)) = f(t) \quad \text{in } V' \text{ a.e.}, \\
p(T) = - \sigma_T S u(T) \quad \text{in } H,
\end{equation}

to its unique solution.

**Definition and Remark 5.5.** Using the previous Definition 5.4 and $Tu = Su + \hat{y}$, we split the right-hand side of the adjoint equation (AE) in two parts and define

\[
\hat{p} := R_1 \left( \sigma_Q(y_Q - \hat{y}) + c \left( y_a - \hat{y} + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( \hat{y} - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \right) \in W(0,T)
\]

and $A : U \to W(0,T)$ with

\[
Au := R_2(-\sigma_Q Su - cSu\chi_{Q_a} - cSu\chi_{Q_b}).
\]

We notice that $A$ is linear and continuous. Furthermore, it holds that

\[
p = Au + \hat{p} = R_2(-\sigma_Q Su - cSu\chi_{Q_a} - cSu\chi_{Q_b}) + R_1 \left( \sigma_Q(y_Q - \hat{y}) + c \left( y_a - \hat{y} + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( \hat{y} - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \right)
\]

solves the adjoint equation (AE) for a given $u \in U_{ad}$.

**Lemma 5.6.** Let $u, v \in U$ and $y := Su, w := \sigma_Q Sv + cSv\chi_{Q_a} + cSv\chi_{Q_b}, \ p := Av \in W(0,T)$. Then, it holds

\[
\int_0^T \langle (Bu)(t), p(t) \rangle_{V',V} \, dt = - \int_0^T \langle w(t), y(t) \rangle_H \, dt - \sigma_T \langle Sv(T), y(T) \rangle_H.
\]

**Proof.** If $Bu$ is the only right-hand side of the weak formulation (SE wt), we have $y = Su$ as the solution. Since $p(t) \in V$ holds, we obtain

\[
\int_0^T \langle (Bu)(t), p(t) \rangle_{V',V} \, dt = \int_0^T \langle y(t), p(t) \rangle_{V',V} + a(t; y(t), p(t)) \, dt.
\]

Integration by parts, using $y(0) = 0$ and $p(T) = - \sigma_T Sv(T)$ yield

\[
\int_0^T \langle y(t), p(t) \rangle_{V',V} + a(t; y(t), p(t)) \, dt = \int_0^T \langle -p_t(t), y(t) \rangle_{V',V} + a(t; y(t), p(t)) \, dt \\
+ \langle p(T), y(T) \rangle_H - \langle p(0), y(0) \rangle_H \\
= \int_0^T \langle -p_t(t), y(t) \rangle_{V',V} + a(t; y(t), p(t)) \, dt \\
- \sigma_T \langle Sv(T), y(T) \rangle_H.
\]
Moreover, it holds that \( y(t) \in V \) and \( p \) solves (5.11) with right-side \(-w\) which means
\[
\int_0^T \langle -p(t), y(t) \rangle_{V', V} + a(t; y(t), p(t)) \, dt = - \int_0^T \langle w(t), y(t) \rangle_H \, dt.
\]
Together we get
\[
\int_0^T \langle (Bu)(t), p(t) \rangle_{V', V} \, dt = - \int_0^T \langle w(t), y(t) \rangle_H \, dt - \sigma_T \langle Sv(T), y(T) \rangle_H.
\]

**Lemma 5.7.** For \( u, v \in U \), it holds
\[
\langle B^*Av, u \rangle_U + \sigma_T \langle Sv(T), Su(T) \rangle_H = - \langle S^*S(\sigma_Q + c\chi_{Q_a} + c\chi_{Q_b})v, u \rangle_U
\]
and
\[
\langle B^*\hat{p}, u \rangle_U - \sigma_T \langle yt - \hat{y}(T), Su(T) \rangle_H = \left\langle S^* \left( \sigma_Q(y_Q - \hat{y}) + c \left( y_a - \hat{y} + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( \hat{y} - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \right), u \right\rangle_U.
\]

**Proof.** Let \( u, v \in U \) and \( y, w, v, p \) be like in the proof of Lemma 5.6. Applying Lemma 5.6, it holds
\[
\langle S^*S(\sigma_Q + c\chi_{Q_a} + c\chi_{Q_b})v, u \rangle_U = \langle S^*w, u \rangle_U = \langle w, Su \rangle_{L^2(0,T;H)} = \int_0^T \langle w(t), y(t) \rangle_H \, dt
\]
\[
= - \int_0^T \langle (Bu)(t), p(t) \rangle_{V', V} \, dt - \sigma_T \langle Sv(T), y(T) \rangle_H
\]
\[
= - \langle Bu, p \rangle_{L^2(0,T;V'), L^2(0,T; V)} - \sigma_T \langle Sv(T), y(T) \rangle_H
\]
\[
= - \langle u, B^*p \rangle_U - \sigma_T \langle Sv(T), y(T) \rangle_H
\]
\[
= - \langle B^*Av, u \rangle_U - \sigma_T \langle Sv(T), y(T) \rangle_H
\]
Up next, using the definition of \( \hat{p} \) and integration by parts, we conclude
\[
\left\langle S^* \left( \sigma_Q(y_Q - \hat{y}) + c \left( y_a - \hat{y} + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( \hat{y} - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \right), u \right\rangle_U
\]
\[
= \int_0^T \left\langle S^* \left( \sigma_Q(y_Q - \hat{y}) + c \left( y_a - \hat{y} + \frac{\mu_a}{c} \right) \chi_{Q_a} - c \left( \hat{y} - y_b + \frac{\mu_b}{c} \right) \chi_{Q_b} \right), Su \right\rangle_H \, dt
\]
\[
= \int_0^T \langle -\hat{p}(t), y(t) \rangle_H + a(t; y(t), \hat{p}(t)) \, dt
\]
\[
= \int_0^T \langle y(t), \hat{p}(t) \rangle_H + a(t; y(t), \hat{p}(t)) \, dt - \sigma_T \langle yt - \hat{y}(T), y(T) \rangle_H
\]
\[
= \int_0^T \langle (Bu)(t), \hat{p}(t) \rangle_{V', V} \, dt - \sigma_T \langle yt - \hat{y}(T), y(T) \rangle_H
\]

28
\[
\langle B^* \hat{p}, u \rangle_U - \sigma_T (y_T - \hat{y}(T), y(T))_H \\
= \langle B^* \hat{p}, u \rangle_U - \sigma_T (y_T - \hat{y}(T), Su(T))_H
\]

where \( y = Su \) is the solution to (5.10) with right-hand side \( Bu \).

**Corollary 5.8.** Let \( p = Au + \hat{p} \) be the solution of the adjoint equation (AE) with end value \( p(T) = \sigma_T (y_T - Tu(T)) \) and let \( y = Tu = Su + \hat{y} \) be the solution of the weak form of the state equation (SE_{wf}) for a given \( u \in U \). Then, for \( u^\delta \in U \), Lemma 5.7 implies

\[
\left\langle S^* \left( \sigma_Q (y_Q - Tu) + c \left( y_a - Tu + \frac{\mu_a}{c} \right) \chi Q_a - c \left( Tu - y_b + \frac{\mu_b}{c} \right) \chi Q_b \right), u^\delta \right\rangle_U \\
= \langle B^* \hat{p}, u^\delta \rangle_U + \langle B^* Au, u^\delta \rangle_U - \sigma_T (y_T - \hat{y}(T), Su^\delta(T))_H + \sigma_T (Su(T), Su^\delta(T))_H \\
= \langle B^* (Au + \hat{p}), u^\delta \rangle_U - \sigma_T (y_T - y(T), Su^\delta(T))_H \\
= \langle B^* p, u^\delta \rangle_U - \langle p(T), Su^\delta(T) \rangle_H.
\]

**Lemma 5.9.** Let \( p = Au + \hat{p} \) be the solution of the adjoint equation (AE) with end value \( p(T) = \sigma_T (y_T - Tu(T)) \) and let \( y = Tu = Su + \hat{y} \) be the solution of the weak form of the state equation (SE_{wf}) for a given \( u \in U \). We can write the gradient of \( \hat{L}_c \) in the form

\[
\frac{\partial \hat{L}_c}{\partial u}(u, \mu_a, \mu_b) u^\delta = \left\langle -B^* p + \begin{pmatrix} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{pmatrix}, u^\delta \right\rangle_U,
\]

where \( B^* : L^2(0, T; V) \rightarrow U \) is the adjoint operator of \( B \) given by

\[
B^* v(t) = \begin{pmatrix} 
\gamma_c \int_{\Gamma_c} b_1(x)v(t, x) \, dA(x) \\
\vdots \\
\gamma_c \int_{\Gamma_c} b_m(x)v(t, x) \, dA(x)
\end{pmatrix}.
\]

**Proof.** Plugging the results of Corollary 5.8 in (5.8) and using \( p(T) = \sigma_T (y_T - Tu(T)) \), we get the representation

\[
\frac{\partial \hat{L}_c}{\partial u}(u, \mu_a, \mu_b) u^\delta = \left\langle S^* \left( \sigma_Q (y - y_Q) + c \max \left\{ 0, Tu - y_b + \frac{\mu_b}{c} \right\} \right) \\
- c \max \left\{ 0, y_a - Tu + \frac{\mu_a}{c} \right\} + \begin{pmatrix} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{pmatrix}, u^\delta \right\rangle_U \\
+ \langle \sigma_T (Tu(T) - y_T), Su^\delta(T) \rangle_H \\
= \left\langle -B^* p + \begin{pmatrix} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{pmatrix}, u^\delta \right\rangle_U - \langle p(T), Su^\delta(T) \rangle_H + \langle p(T), Su^\delta(T) \rangle_H
\]

29
\[
- B^* p + \begin{pmatrix} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{pmatrix} , \ u^\delta \right\rangle_U .
\]

It remains to determine $B^*$. Let $u \in U$ and $v \in L^2(0, T; V)$ be arbitrary. Then, it holds
\[
\int_0^T \langle (Bu)(t), v(t) \rangle_{V', V} dt = \gamma_c \int_0^T \sum_{i=1}^m u_i(t) \int_{\Gamma_c} b_i(x) v(t) \, dA(x) \, dt
\]
\[
= \langle u, B^* v \rangle_U
\]

with
\[
B^* v(t) = \begin{pmatrix} \gamma_c \int_{\Gamma_c} b_1(x) v(t, x) \, dA(x) \\ \vdots \\ \gamma_c \int_{\Gamma_c} b_m(x) v(t, x) \, dA(x) \end{pmatrix}.
\]

Since we are able to solve the adjoint equation (AE), we obtain $p = Au + \hat{p}$ for a given control $u \in U$. Hence, we can numerically deal with $\frac{\partial L_c}{\partial u}$. 

30
Numerical Consideration of the Optimal Control Problem

In the previous chapters, we made the theoretical preparations for the practical realization of the optimal control problem. Now, in this chapter, we show numerical results. Using finite elements and the Galerkin ansatz, we convert the state and the adjoint equation into ordinary differential equations. To obtain the solution of the arising ordinary differential equations, we apply the implicit Euler method. Moreover, to solve the minimization problem within the augmented Lagrangian method, we apply the gradient projection method, a first-order method. To recall, we want to solve the problem

$$\min \ J(y,u) = \frac{\sigma_Q}{2} \int_0^T \|y(t) - y_Q(t)\|^2_H dt + \frac{\sigma_T}{2} \|y(T) - y_T\|^2_H + \frac{1}{2} \sum_{i=1}^m \sigma_i \|u_i\|^2_{L^2(0,T)}$$

subject to the state equation

$$y_t(t,x) - \lambda \Delta y(t,x) + v(t,x) \cdot \nabla y(t,x) = f(t,x) \quad \text{in } Q = (0,T) \times \Omega \ \text{a.e.},$$

$$\lambda \frac{\partial y}{\partial \eta}(t,x) + \gamma_c y(t,x) = \gamma_c \sum_{i=1}^m u_i(t) b_i(x) \quad \text{in } \Sigma_c = (0,T) \times \Gamma_c \ \text{a.e.},$$

$$\lambda \frac{\partial y}{\partial \eta}(t,x) + \gamma_{out} y(t,x) = \gamma_{out} y_{out}(t) \quad \text{in } \Sigma_{out} = (0,T) \times \Gamma_{out} \ \text{a.e.},$$

$$y(0,x) = y_0(x) \quad \text{in } \Omega \ \text{a.e.}$$

as well as

$$u \in U_{ad} \quad \text{and} \quad y_a \leq Tu \leq y_b \text{ in } Q \ \text{a.e.}$$

For the numerical treatment and all the tests, we choose the unit square

$$\Omega := (0,1) \times (0,1)$$

as our domain $\Omega$ and consequently, as the representative for our room. Furthermore, we set the final time $T = 1$.

6.1 || Solving the State and the Adjoint Equation

In the previous chapters, we dealt with two partial differential equations as part of the augmented Lagrangian method. In the following, we solve the two equations numerically. For simplicity reasons, we rewrite the weak form of both equations. For the state equation (SE) we have

$$\int_{\Omega} y_t(t,x) \varphi(x) \, dx + a(t;y(t,x),\varphi(x)) = g(t,\varphi(x)) \quad \text{for } \varphi \in V = H^1(\Omega),$$

$$y(0) = y_0 \quad \text{in } H = L^2(\Omega).$$
Besides, for the adjoint equation (AE), for \( \varphi \in V = H^1(\Omega) \), it holds
\[
\int_{\Omega} -p_t(t, x)\varphi(x) \, dx + a(t; \varphi(x), p(t, x)) = \int_{\Omega} \sigma_Q(y_Q(t, x) - T u(t))\varphi(x) \\
+ c \left( y_a(t, x) - T u(t) + \frac{\mu_a(t, x)}{c} \right) \chi_{Qa}(x) \\
- c \left( T u(t) - y_b(t, x) + \frac{\mu_b(t, x)}{c} \right) \chi_{Qb}(x) \, dx,
\]
\( p(T) = \sigma_T(y_T - T u(T)) \) in \( H = L^2(\Omega) \).

The state and the adjoint equation contain the terms
\[
a(t; \psi, \varphi) = \lambda \int_{\Omega} \nabla \psi(x) \cdot \nabla \varphi(x) \, dx + \gamma_c \int_{\Gamma_c} \psi(x)\varphi(x) \, dA(x) + \gamma_{out} \int_{\Gamma_{out}} \psi(x)\varphi(x) \, dA(x) \\
+ \int_{\Omega} v(t, x) \cdot \nabla \psi(x)\varphi(x) \, dx
\]
and
\[
g(t, \varphi) = \int_{\Omega} f(t, x)\varphi(x) \, dx + \gamma_c \sum_{i=1}^{m} u_i(t) \int_{\Gamma_c} b_i(x)\varphi(x) \, dA(x) + \gamma_{out} y_{out}(t) \int_{\Gamma_{out}} \varphi(x) \, dA(x).
\]

### 6.1.1 Galerkin Ansatz and Implicit Euler Method for the State Equation

Given a triangulation \( \Delta \) of \( \Omega \), where \( \partial \Delta_c \) and \( \partial \Delta_{out} \) belong to the boundary \( \Gamma_c \) and \( \Gamma_{out} \), respectively, the finite element space on \( \Delta \) is denoted as \( V_{\Delta} \). Let \( V_{\Delta} \subset V \) be spanned by the piecewise linear functions \( \varphi_1, ..., \varphi_N \) with \( N \in \mathbb{N} \) where it holds that
\[
\varphi_i(v_j) = \begin{cases} 
1 & \text{for } i = j, \\
0 & \text{for } i \neq j,
\end{cases} \quad \text{for } 1 \leq i, j \leq N,
\]
with \( v_j \) as the \( j \)-th node of the triangulation \( \Delta \). In this way, involving the Galerkin ansatz \( y(t, x) \approx \sum_{i=1}^{N} \tilde{y}_i(t)\varphi_i(x) \) for suitable \( \tilde{y}_1, ..., \tilde{y}_N \) (see [ZTZ05, Chapter 3] for instance), we approximate
\[
\int_{\Omega} y_i(t, x)\varphi(x) \, dx + a(t; y(t, x), \varphi(x)) = g(t, \varphi(x)) \quad \text{for } \varphi \in V = H^1(\Omega)
\]
by
\[
\sum_{i=1}^{N} \tilde{y}_i(t) \int_{\Omega} \varphi_i(x)\varphi_j(x) \, dx + \sum_{i=1}^{N} \tilde{y}_i(t)a(t; \varphi_i(x), \varphi_j(x)) = g(t, \varphi_j(x)) \quad \text{for } j = 1, ..., N.
\]
Before we solve this semidiscrete system of ordinary differential equations in time with the unknown \( \tilde{y} = (\tilde{y}_1, ..., \tilde{y}_N) \), we give the resulting matrices and explain how to calculate
them. At first, there are the entries
\[ M_{i,j} := \int_{\Omega} \varphi_i(x) \varphi_j(x) \, dx \approx \sum_{\tau \in \Delta} \int_{\tau} \varphi_i(x) \varphi_j(x) \, dx. \]
of the mass matrix \( M \). Moreover, inside the bilinear form \( a \) and the right-hand side \( g \), we have the terms
\[ A_{i,j}^1 := \int_{\Omega} \nabla \varphi_i(x) \nabla \varphi_j(x) \, dx \approx \sum_{\tau \in \Delta} \int_{\tau} \nabla \varphi_i(x) \nabla \varphi_j(x) \, dx, \]
\[ A_{i,j}^2 := \int_{\Gamma_c} \varphi_i(x) \varphi_j(x) \, dA(x) \approx \sum_{\tau \in \partial \Delta_c} \int_{\tau} \varphi_i(x) \varphi_j(x) \, dA(x), \]
\[ A_{i,j}^3 := \int_{\Gamma_{out}} \varphi_i(x) \varphi_j(x) \, dA(x) \approx \sum_{\tau \in \partial \Delta_{out}} \int_{\tau} \varphi_i(x) \varphi_j(x) \, dA(x), \]
\[ A_{i,j}^4(t) := \int_{\Omega} v(t, x) \cdot \nabla \varphi_i(x) \varphi_j(x) \, dx \approx \sum_{\tau \in \Delta} \int_{\tau} v(t, x) \cdot \nabla \varphi_i(x) \varphi_j(x) \, dx, \]
\[ G_{j}^1(t) := \int_{\Omega} f(t, x) \varphi_j(x) \, dx \approx \sum_{\tau \in \Delta} \int_{\tau} f(t, x) \varphi_j(x) \, dx, \]
\[ G_{k,j}^2 := \int_{\Gamma_c} b_k(x) \varphi_j(x) \, dA(x) \approx \sum_{\tau \in \partial \Delta_c} \int_{\tau} b_k(x) \varphi_j(x) \, dA(x) \quad (k \in \{1, \ldots, m\}), \]
and
\[ G_{j}^3 := \int_{\Gamma_{out}} \varphi_j(x) \, dA(x) \approx \sum_{\tau \in \partial \Delta_{out}} \int_{\tau} \varphi_j(x) \, dA(x). \]

In this thesis, we choose the linear finite element space
\[ V_\Delta := \{ \varphi \in C(\bar{\Omega}) : \varphi|_{\tau} \in \mathcal{P}_1 \text{ for all } \tau \in \Delta \} \]
where \( \mathcal{P}_1 := \{ f(x) = a_1 x + a_0 \mid a_1, a_0 \in \mathbb{R} \} \) describes the space of linear polynomials. To get values for every entry of the matrices, we use the reference triangle or quadrature formulas. We show this exemplary for the entries
\[ \sum_{\tau \in \Delta} \int_{\tau} \varphi_i(x) \varphi_j(x) \, dx \]
of the mass matrix \( M \). For each simplex \( \tau \), we define the local mass matrix \( M|_{\tau} \) as
\[ (M|_{\tau})_{i, j} := \int_{\tau} \varphi_{i_{\tau}}(x) \varphi_{j_{\tau}}(x) \, dx \]
for \( i_{\tau}, j_{\tau} \in \{1, 2, 3\} \). Restricted to one simplex, the basis \( \varphi_i \) is identical to the local one \( \varphi_{i_{\tau}} \). Hence, we determine all \( 3 \times 3 \) local mass matrices \( M|_{\tau} \) and sum each local contribution into the global one. Thereto, we use a reference triangle \( \hat{\tau} \) which is a triangle with vertices
\[ \hat{v}_1 = (0, 0), \hat{v}_1 = (1, 0), \hat{v}_1 = (0, 1). \] Every triangle \( \tau \) of vertices \( v_i \) is supposed to be the image of \( \hat{\tau} \) under the affine mapping \( R: \hat{\tau} \rightarrow \tau \) such that

\[ R(\hat{v}_i) = v_i = (x_i, y_i), \quad i \in \{1, 2, 3\}, \]

with

\[ R(\hat{p}) = \Lambda^T \hat{p} + r \]

where \( \hat{p} = (\hat{x}, \hat{y}) \) represents the coordinates of a point in the reference system of the reference triangle, \( r = (x_1, y_1)^T \)

\[ \Lambda = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{pmatrix}. \]

Figure 2: The mapping \( R \) transforms the reference triangle \( \hat{\tau} \) into \( \tau \)

This yields

\[ (M|_\tau)_{i,j} = \int_{\hat{\tau}} \varphi_i,(x,y) \varphi_j,(x,y) \, dx \, dy = |\det(\Lambda)| \int_{\hat{\tau}} \varphi_i,(R(\hat{x}, \hat{y})) \varphi_j,(R(\hat{x}, \hat{y})) \, d\hat{x} \, d\hat{y}. \]

In the last step, we use a Gaussian quadrature rule with one node to obtain

\[ (M|_\tau)_{i,j} = |\det(\Lambda)| \int_{\hat{\tau}} \varphi_i,(R(\hat{x}, \hat{y})) \varphi_j,(R(\hat{x}, \hat{y})) \, d\hat{x} \, d\hat{y} \]

\[ = \frac{1}{2} |\det(\Lambda)| \varphi_i,(R(\xi, \eta)) \varphi_j,(R(\xi, \eta)). \]

with \( (\xi, \eta) = (\frac{1}{3}, \frac{1}{3}) \). What we get at the end, is an approximation for all entries of the single matrices or vectors which build the ordinary differential equation

(6.2) \[ \dot{\tilde{y}}(t) + A(t)\tilde{y}(t) = G(t) \]

(6.3) \[ \tilde{y}(0) = \tilde{y}^0 \]

with the unknown \( \tilde{y} \) where the matrices \( \tilde{M}, A(t) \in \mathbb{R}^{N \times N} \) and the time-dependent right-hand side vector \( G(t) \in \mathbb{R}^N \) are given by

\[ \tilde{M}_{i,j} \approx M_{i,j}, \]

\[ A_{i,j}(t) \approx \lambda A^1_{i,j} + \gamma_c A^2_{i,j} + \gamma_{\text{out}} A^3_{i,j} + A^4_{i,j}(t), \]

\[ G_j(t) \approx G^1_j(t) + \gamma_c \sum_{k=1}^m u_k(t) G^2_{k,j}(t) + \gamma_{\text{out}} y_{\text{out}}(t) G^3_j. \]
for $i, j \in \{1, \ldots, N\}$. Due to $y_0$, the initial value $\tilde{y}^0 \in \mathbb{R}^N$ can be determined by

$$y_0 \approx \sum_{i=1}^{N} \tilde{y}_i^0 \varphi_i$$

with the aid of (6.1).

It remains to solve the ordinary differential equations (6.2) with initial condition (6.3). Therefore, we apply the implicit Euler method. In the following part, we introduce the method for a general ordinary differential equation

$$\dot{\tilde{y}} = f(t, \tilde{y}), \quad \tilde{y}(t_0) = \tilde{y}^0$$

with Lipschitz-continuous right-hand side $f$. Choosing a discretization step size $h > 0$ and considering the discrete time steps $t_k = t_0 + kh$ for $k = 1, 2, \ldots, N_t$ ($N_t \in \mathbb{N}$), the method produces

$$\tilde{y}^{k+1} = \tilde{y}^k + hf(t_{k+1}, \tilde{y}^{k+1}) \quad \text{for } k \in \{0, 1, 2, \ldots, N_t - 1\}$$

where we set $\tilde{y}^k := \tilde{y}(t_k)$. In our case, for every time step starting from $t_0 = 0$, we have to solve the linear system

$$(M + hA(t_{k+1}))\tilde{y}^{k+1} = M\tilde{y}^k + hG(t_{k+1}) \in \mathbb{R}^N.$$ 

The implicit Euler method, which is a first-order method, is chosen because it does not have any stability restrictions with respect to the time steps.

### 6.1.2 Galerkin Ansatz and Implicit Euler Method for the Adjoint Equation

Analogously, we solve the adjoint equation. One should note that the implicit Euler method is thereby utilized backwards in time. At the end, we have to solve

(6.4)

$$(M + hA(t_k)^T)p^k = M\tilde{y}^{k+1} + h\left(\sigma_Q(y_Q(t_k) - \tilde{y}^k) + c \left(y_a(t_k) - \tilde{y}^k + \frac{\mu_a(t_k)}{c}\right)\chi_Qa - c \left(y_b(t_k) + \frac{\mu_b(t_k)}{c}\right)\chi_Qb\right)$$

for $k \in \{N_t - 1, N_t - 2, \ldots, 0\}$

$$\tilde{p}(N_t) = \sigma_T(y_T - \tilde{y}^{N_t})$$

for the unknown $\tilde{p} \in \mathbb{R}^N$. We note that $\tilde{y}$ is the solution of the state equation from the foregoing section.

### 6.2 || The Practicable Code

Numerically, we can solve our problem, which is a restricted optimization problem and an optimal control problem at the same time, with the augmented Lagrangian method. We already presented a pseudo code for this method in Algorithm 3. When we have a
look inside the code of Algorithm 3, we find that every iteration contains minimizing the function

$$\hat{L}_c(u, \mu_a, \mu_b) = \hat{J}(u) + \frac{c}{2} \left( \max \left\{ 0, y_a - Tu + \frac{\mu_a}{c} \right\} \right)^2_{L^2(Q)}$$

$$+ \left( \max \left\{ 0, Tu - y_b + \frac{\mu_b}{c} \right\} \right)^2_{L^2(Q)} - \frac{1}{2c} \left( \|\mu_a\|^2_{L^2(Q)} + \|\mu_b\|^2_{L^2(Q)} \right).$$

For solving this minimization problem, we use the first-order

Algorithm 4 : Gradient projection method (for $f \in C^1$)

1: Data: Initial guess $u^0$, projection function $P$
2: begin
3: set $k = 0$;
4: while no convergence and $k < k_{\text{max}}$ do
5: set descent direction $d^k = -\nabla f(u^k)$;
6: choose step size $s_k \in (0, 1)$;
7: calculate $u^{k+1} = P(u^k + s_k d^k)$;
8: set $k = k + 1$;
9: end.

A detailed description of this algorithm can be found in [Kel99, Chapter 5.4]. We choose the step size by means of the sufficient decrease condition for line searches (compare with 3.2)

$$f(u^{k+1}) - f(u^k) \leq -\alpha s_k \|u^{k+1} - u^k\|^2$$

where $\alpha$ is typically set to $10^{-4}$. We choose $\zeta \in (0, 1)$ and try to find the least integer $m$ such that $s_k = \zeta^m$ fulfills condition (6.5). The projection function $P : U \to U_{\text{ad}}$ is the orthogonal projection to the control constraints’ set $U_{\text{ad}}$, i.e. if we have $u < u_a$ or $u > u_b$, $P$ projects $u$ to $u_a$ or $u_b$, respectively. In this way, we guarantee that the control bounds are satisfied.

The gradient projection method requires the gradient of the function which shall be minimized. In line with this, we remember the gradient of the augmented Lagrange function

$$\frac{\partial \hat{L}_c}{\partial u}(u, \mu_a, \mu_b) u^\delta = \left\langle -B^* p + \begin{pmatrix} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{pmatrix}, u^\delta \right\rangle_U,$$

specified in Lemma 5.9, where $p$ is the solution of the adjoint equation (AE).

After combining Algorithm 3 with Algorithm 4, we present a more practicable algorithm recorded in
Algorithm 5: Gradient projection augmented Lagrangian method

1: Data: \((\mu^0_a, \mu^0_b), u^{0,0}, c^0 > 0, \beta > 1, \varepsilon, \varepsilon_{gp}\);
2: begin
3: set \(n, k = 0\), choose \(\hat{u}, \hat{u}\) large enough to enter the loops;
4: solve the state equation for \(u^{n,k}\) to get \(T u^{n,k}\);
5: solve the adjoint equation to get \(p^{n,k}\) (needed for (6.6));
6: while \(\|\hat{u} - u^{n,k}\| > \varepsilon\) and \(n < n_{\text{max}}\) do
7: set \(\hat{u} = u^{n,k}, c = c^n\);
8: while \(\|\hat{u} - u^{n,k}\| > \varepsilon_{gp}\) and \(k < k_{\text{max}}\) do
9: set \(\hat{u} = u^{n,k}\);
10: % Step 5 from Algorithm 3 turns into:
11: calculate descent direction \(d^k = -\frac{\partial L_c}{\partial u}(u^{n,k}, \mu^n_a, \mu^n_b)\) using \(p^{n,k}\);
12: choose step size \(s_k \in (0, 1)\) using (6.5);
13: do one gradient step to get \(u^{n,k+1} = P(u^{n,k} + s_k d^k)\);
14: solve the state equation for \(u^{n,k+1}\) to get \(T u^{n,k+1}\);
15: solve the adjoint equation to get \(p^{n,k+1}\) (needed for (6.6));
16: set \(k = k + 1\);
17: set \(u^{n+1,0} = u^{n,k}\);
18: update the Lagrange multipliers via (5.6) and (5.7) to get \(\mu^{n+1}_a, \mu^{n+1}_b\);
19: update \(c^{n+1} = \beta c^n\);
20: set \(n = n + 1, k = 0\), choose \(\hat{u}\) high enough to enter the inner loop again;
21: end.

6.3 Tests

All tests in this section are implemented on a Notebook Acer Aspire E15 E5-71G-717X, Intel® Core™ i7-5500U 2.4GHz (with Turbo Boost up to 3.0GHz) and 8GB RAM in the programming language MATLAB.

In the following we call the inner loop of Algorithm 5 the gradient loop and the outer loop the Lagrangian loop. For Test 1, we look at different discretizations for the arising equations using linear finite elements with \(N^1_x = 185\) nodes, \(N^2_x = 320\) nodes and \(N^3_x = 712\). The time interval is divided into \(N_t = 101\) equidistantly distributed time steps. To create the geometry and the mesh, we employ the MATLAB routines createpde, geometryFromEdges and generateMesh. For solving the linear systems, MATLAB’s backslash command is utilized. In Figure 3, we can see the different generated meshes.

![Figure 3: Meshes of Ω with \(N^1_x\), \(N^2_x\) and \(N^3_x\) nodes from left to right, respectively](image-url)
Furthermore, we note that the operator $B^\star$, which is given by the representation in Lemma 5.9, can be obtained by transposing the matrix $\gamma c G^2$ appearing in the right-hand side $G$ of the ordinary differential equation (6.2).

In our tests, we basically choose the same prescribed functions and parameters than Luca Mechelli and Stefan Volkwein employ in their tests in Chapter 6 of the paper "POD-Based Economic Optimal Control of Heat-Convection Phenomena" (see [MV18]). The right-hand side $f$ of (SEa) is fixed to be zero. We have four controls with shape functions

$$b_1(x) = \begin{cases} 1 & \text{if } x_1 = 0, 0 \leq x_2 \leq 0.25, \\ 0 & \text{otherwise,} \end{cases}$$

$$b_2(x) = \begin{cases} 1 & \text{if } 0.25 \leq x_1 \leq 0.5, x_2 = 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$b_3(x) = \begin{cases} 1 & \text{if } x_1 = 1, 0.5 \leq x_2 \leq 0.75, \\ 0 & \text{otherwise,} \end{cases}$$

$$b_4(x) = \begin{cases} 1 & \text{if } 0.5 \leq x_1 \leq 0.75, x_2 = 0, \\ 0 & \text{otherwise.} \end{cases}$$

The physical parameters are $\lambda = 1.0$, $\gamma c = 1.0$ and $\gamma_{out} = 0.03$. We set $y_0 = |\sin(2\pi x_1)\cos(2\pi x_2)|$ ($x = (x_1, x_2) \in \Omega$) as the initial condition of the state equation. The entries of the velocity field

$$v(t, x) = (v_1(t, x), v_2(t, x)) \ (t \in [0, T])$$

are given by

$$v_1(t, x) = \begin{cases} -1.6 & \text{if } t < 0.5, x \in V_1, \\ -0.6 & \text{if } t \geq 0.5, x \in V_2, \\ 0 & \text{otherwise,} \end{cases}$$

$$v_2(t, x) = \begin{cases} 0.5 & \text{if } t < 0.5, x \in V_1, \\ 1.5 & \text{if } t \geq 0.5, x \in V_2, \\ 0 & \text{otherwise,} \end{cases}$$

where we have

$$V_1 = \{x = (x_1, x_2) \in \Omega \mid 4x_1 + 12x_2 \geq 3, 4x_1 + 12x_2 \leq 13\},$$

$$V_2 = \{x = (x_1, x_2) \in \Omega \mid x_1 + x_2 \geq 0.5, x_1 + x_2 \leq 1.5\}.$$

Figure 4: The shape functions (blue) and the region of the velocity field (red) in the domain $\Omega$
Furthermore, we define the outside temperature
\[
y_{\text{out}}(t) = \begin{cases} 
-1 & \text{if } 0 \leq t < 0.5, \\
1 & \text{if } 0.5 \leq t \leq T,
\end{cases}
\]
and the targets
\[
y_Q(t, x) = \min(2.0 + t, 3.0) \quad \text{as well as} \quad y_T(x) = y_Q(T, x).
\]
The state constraints are fixed as
\[
y_a(t) = 0.5 + \min(2t, 2.0) \quad \text{and} \quad y_b = 3.0.
\]
On the one hand, our state variable \(y\) should track the temperature profile \(y_Q\) during our optimization procedure as accurately as possible. On the other hand, \(y\) is not allowed to be lower than \(y_a\) and bigger than \(y_b\). Regarding the next figure, \(y\) has to be always between the green and red line.

![Figure 5: The temperature profile \(y_Q\) and the state constraints \(y_a, y_b\)](image)

We restrict our control by setting \(u_a(t) = 0\) and \(u_b(t) = 7\) for \(t \in [0, T]\).

### 6.3.1 Test 1

Test 1 deals with exactly the same functions and initial guesses than Test 1 in [MV18]: Let \(m = 4\). As starting control we set
\[
u_i^0(t) = 3.5 \quad \text{for } t \in [0, T] \text{ and } i \in \{1, \ldots, m\}.
\]
The cost functional (5.1) weights are \(\sigma_T = \sigma_Q = 0\) and \(\sigma_i = 1.0\) for \(i \in \{1, \ldots, m\}\). In order to solve the partial differential equations (SE) and (AE), we use the implicit Euler method with equidistant time step \(\Delta t = 0.01\) as mentioned in the Sections 6.1.1 and 6.1.2. We start our algorithm with
\[
\mu_a^0 = \max\{0, c_0(y_a - Tu_0^0)\}, \\
\mu_b^0 = \max\{0, c_0(Tu_0^0 - y_b)\}
\]
and set $\varepsilon = 10^{-5}$ and $\varepsilon_{gp} = 0.2\varepsilon$ as the stopping criterion. For finding the step size by (6.5), we start with $\zeta = 0.8$. Results for different settings can be found in Table 2.

<table>
<thead>
<tr>
<th>$N_x$</th>
<th>$c$</th>
<th>$\beta$</th>
<th>Iterations (Gradient loops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>185</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>185</td>
<td>1</td>
<td>190</td>
</tr>
<tr>
<td>3</td>
<td>185</td>
<td>0.1</td>
<td>190</td>
</tr>
<tr>
<td>4</td>
<td>185</td>
<td>0.1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>185</td>
<td>40</td>
<td>1.2</td>
</tr>
<tr>
<td>6</td>
<td>185</td>
<td>50</td>
<td>1.2</td>
</tr>
<tr>
<td>7</td>
<td>320</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>320</td>
<td>1</td>
<td>350</td>
</tr>
<tr>
<td>9</td>
<td>320</td>
<td>0.1</td>
<td>350</td>
</tr>
<tr>
<td>10</td>
<td>320</td>
<td>0.1</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>320</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>320</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>712</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>712</td>
<td>1</td>
<td>350</td>
</tr>
<tr>
<td>15</td>
<td>712</td>
<td>0.01</td>
<td>350</td>
</tr>
<tr>
<td>16</td>
<td>712</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>712</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>18</td>
<td>712</td>
<td>350</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 2: Results for the method with different settings for Test 1

We note that the number of gradient loops is the number of iterations for the Lagrangian loop. Depending on the $c$ and $\beta$ combination, the algorithm executes more than two Lagrangian loop steps or not. When we have a look at the plots of the control for different settings, we see problems.
Figure 6: The control $u_i, i \in \{1, 2, 3, 4\}$, for different settings (First row: $N_x = 185$, second row: $N_x = 320$, third row: $N_x = 712$)

The algorithm does not perform the anticipated results. Either it stops too soon or the control value at the time $T$ is not equal to zero. However, the value has to be zero since the gradient $\frac{\partial \hat{L}_c}{\partial u}$ tends to zero in theory and so (6.6) and $p(T) = 0$ yield

$$ u(T) = -B^*0 + u(T) = -B^*p(T) + \begin{pmatrix} \sigma_1 u_1 \\ \sigma_2 u_2 \\ \vdots \\ \sigma_m u_m \end{pmatrix} (T) = \frac{\partial \hat{L}_c}{\partial u} (u(T)) \to 0 $$

for the last computed control at ending time $T$. The reason behind the problem $u(T) \neq 0$ in the algorithm is that (6.5) permanently produces very little step sizes. In Table 2 we can see that for none of our discretizations, $c = 1$ let the algorithm run for more than one Lagrangian loop - no matter what $\beta$ we choose. Considering $N_x = 712$, either we have to start with $c \leq 0.01$ in combination with $\beta \geq 350$. But then, $\beta$ is very big for the total procedure, which implies that the penalty increases too fast, or we initiate with $c \geq 350$ with the result $u(T) \neq 0$ for the computed $u$. What causes this problem? It is obvious that a small $c$ in the beginning, $0.01 < c < 350$ for $N_x = 712$ to be precise, does not affect the multipliers (see (6.8), (6.9)) and the result of the adjoint equation (see (6.4)) enough such that the new $u$ differs sufficiently from the old $u$ in order to stay in the loop. But in contrast, a very small $c$, actually $c \leq 0.01$, shrinks the solution of the adjoint since the right-hand side of (6.4) becomes smaller. Hence, plugging in the small adjoint solution and concerning its consequence of obtaining bigger step sizes resulting from (6.5), the difference between the new $u$ and the old one is big enough to remain in the loops. We observe that $c$ has to be pushed a lot (see Table 3) and kept away from the region, where it does not influence the new $u$ as described above, because otherwise the algorithm stagnates.

To improve the method’s results, we start with a small $c$ and a big $\beta$. After the first iteration, we change the $\beta$ and make it smaller. This means that we start our iteration with a little punishment. Then, we punish a lot in the next step to remain in the loop and in the following steps, we again increase the punishment but in a lower ratio. More precisely, we have a big $\beta_0$ for the first $c$-update and for the outstanding iterations we take $\beta \in [4, 10]$ which is common for Lagrangian methods. This is recorded in Table 3 and the new results can be seen in Figure 7.
After comparing our results for the control with the ones from [MV18], we infer that the setting $N_x = 712, c = 0.01, \beta_0 = 350, \beta = 4$ (Row 7, Table 3) is the most similar. We note that the tests in [MV18] are implemented in the programming language C. The
authors apply a primal-dual active set strategy which can be interpreted as a semismooth Newton method (see [MV18, pages 9-11]). In addition, a reduced-order approach based on proper orthogonal decomposition is used to speed up their method (see [MV18]).

(a) Augmented Lagrangian method results for \( N_x = 712, c = 0.01, \beta_0 = 350, \beta = 4 \)

(b) Primal-dual active set strategy results in [MV18]

Figure 8: Comparison of the control \( u \) for the different methods

We observe a jump of all controls at time \( t = 0.5 \). It is caused by the sudden change of the velocity field at that time step (see (6.7)). In addition, we provide the solutions of the state equation for different time steps for the last computed control \( u \) - compare with Figure 8 (a).

(a) time step 1 of 101
(b) time step 33 of 101
(c) time step 66 of 101
(d) time step 101 of 101

Figure 9: Solutions for the state equation in the setting \( N_x = 712, c = 0.01, \beta_0 = 350, \beta = 4 \) (Row 3 of Table 3) for different time steps
Furthermore, the solutions of the adjoint equation for different time steps for the last computed control \( u \) are allocated.

![Graphs showing solutions for different time steps](image)

Figure 10: Solutions for the adjoint equation in the setting \( N_x = 712, c = 0.01, \beta_0 = 350, \beta = 4 \) (Row 3 of Table 3) for different time steps

### 6.3.2 Test 2

By taking the starting control

\[
u_i^0(t) = 3.5 \quad \text{for } t \in [0,T] \text{ and } i \in \{1, \ldots, m\},
\]

the method needed the adapted \( \epsilon \)-update to produce acceptable results. Now, for Test 2, every function and the initial data except for the initial control \( u^0 \) stays the same. This time, we run our algorithm with

\[
u_i^0(t) = 0 \quad \text{for } t \in [0,T] \text{ and } i \in \{1, \ldots, m\}.
\]

In this way, the control value at time \( T \) will not be equal to zero. The gradient step including (6.6) shows that the controls \( u_i, i \in \{1, \ldots, m\} \), will always be zero at time \( T \) in every step of the algorithm. So, inductively the last computed control is zero at time \( T \). The results are listed in Table 4.
<table>
<thead>
<tr>
<th>$N_x$</th>
<th>$c$</th>
<th>$\beta$</th>
<th>Iterations (Gradient loops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>712</td>
<td>1</td>
<td>4, 4, 507, 61, 269, 166, 133, 425, 51, 419, 3, 4, 2</td>
</tr>
<tr>
<td>2</td>
<td>712</td>
<td>1</td>
<td>6, 4, 582, 3, 78, 171, 722, 348, 2</td>
</tr>
<tr>
<td>3</td>
<td>712</td>
<td>1</td>
<td>8, 4, 515, 3, 2</td>
</tr>
</tbody>
</table>

Table 4: Results for the method with different settings for Test 2

![Graphs](image)

Figure 11: The control $u_i, i \in \{1, 2, 3, 4\}$, for different values of $\beta$ corresponding to Table 4

Table 4 and Figure 11 show that the algorithm for $\beta = 4$ and $\beta = 6$ performs similar results than our adapted method in the test above. In case $\beta = 8$, the method stops too early. Here, we can start with $c = 1$. The new starting control does not exert the adapted $c$-update, i.e. no high increase of $c$ is needed after the first iteration in order to gain expected results. All in all, the algorithm works without the necessity of the adapted $c$-update when we initiate with the new $u^0$.

In fact, when all initial controls $u_i^0, i \in \{1, ..., m\}$, have equal values at every time and these values are lower or equal than 0.323, the algorithm does not demand the adapted $c$-update. All other starting controls - more precisely, when their equal values are greater than 0.323 - require the adapted $c$-update. We note that $u^0$ has to lie in $U_{ad}$. This means that $u_i^0(t) < 0$ or $u_i^0(t) > 7$ is not admissible.

6.3.3 Test 3

In the previous tests, we wanted to find the economic model predictive control (see [GP17, Chapter 8]): Since we chose $\sigma_Q, \sigma_T = 0$, we did not want to reach the temperature targets

$$y_Q(t, x) = \min(2.0 + t, 3.0) \quad \text{and} \quad y_T(x) = y_Q(T, x).$$

Our focus was to fulfill the state constraints and push the controls as low as possible. Now, we activate the temperature targets through setting $\sigma_Q, \sigma_T = 1$. This also means that we have $p(T) = \sigma_T(y_T - y(T))$ instead of $p(T) = 0$ (compare with (AE)). We test again with

$$u_i^0(t) = 0 \quad \text{for} \quad t \in [0, T] \text{ and } i \in \{1, ..., m\}.$$

All unmentioned values are the same than in the other tests, i.e. $\sigma_i = 1$ for $i \in \{1, ..., m\}$, $\Delta t = 0.01$, $\mu_a^0 = \max\{0, c_0(y_a - T u^0)\}$, $\mu_b^0 = \max\{0, c_0(T u^0 - y_b)\}$, $\varepsilon = 10^{-5}$, $\varepsilon_{gp} = 0.2\varepsilon$.
and $\zeta = 0.8$. We try different settings for $\beta$ while $c$ is set to one. The results are recorded in Table 5 and Figure 12.

<table>
<thead>
<tr>
<th>$N_x$</th>
<th>$c$</th>
<th>$\beta$</th>
<th>Iterations (Gradient loops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>712</td>
<td>1 4</td>
<td>{5, 8, 447, 3, 270, 172, 126, 430, 46, 327, 6, 11, 6, 2}</td>
</tr>
<tr>
<td>2</td>
<td>712</td>
<td>1 6</td>
<td>{5, 321, 516, 3, 80, 158, 627, 3, 3, 5, 2}</td>
</tr>
<tr>
<td>3</td>
<td>712</td>
<td>1 8</td>
<td>{5, 513, 472, 3, 2}</td>
</tr>
</tbody>
</table>

Table 5: Results for the method with different settings for Test 3

Figure 12: The control $u_i$, $i \in \{1, 2, 3, 4\}$, for different values of $\beta$ corresponding to Table 5

Figure 13: The differences in the controls $u_i$, $i \in \{1, 2, 3, 4\}$ between $\sigma_Q, \sigma_T = 0$ and $\sigma_Q, \sigma_T = 1$ for the setting $N_x = 712, c = 1$ and $\beta = 4$

If we compare the result for $\sigma_Q, \sigma_T = 0$ with the result for $\sigma_Q, \sigma_T = 1$, i.e. Figure 11 (a) with Figure 12 (a), we can see that the shapes of the graphs are similar. The biggest
difference is that for tracking the temperature profiles $y_Q$ and $y_T$, we obviously have to pay more costs for the controls between the time $t = 0.07$ until time $t = 0.5$. After $t = 0.5$, there is no big difference in the costs: For $\sigma_Q, \sigma_T = 1$, they are higher until time $t = 0.87$. Then, the costs for $\sigma_Q, \sigma_T = 0$ even transcend the costs for $\sigma_Q, \sigma_T = 1$ a little. These statements can be seen in Figure 13.
Chapter 7

CONCLUSION AND OUTLOOK

In this thesis, the application of the augmented Lagrangian method combined with the first-order gradient projection method to the optimal boundary control problem (OCP) governed by the heat-convection phenomena (SE) was investigated. Let us recall the procedure to our results.

In the first place, we introduced the augmented Lagrangian method for equality constrained optimization problems and derived its properties. Advantages compared to the common Lagrangian method were revealed and after presenting the numerical realization of the augmented Lagrangian method, first examples were given to make the reader familiar to the algorithm. How to deal with inequality constraints, which are part of the optimal control problem, was shown in the end of Chapter 3.

We proceeded with facing the convection-diffusion model. This linear parabolic differential equation with Robin boundary conditions is the state constraint of our optimal control problem. To be able to work with the equation theoretically and numerically, we determined its weak formulation followed by the proof of the well-posedness and the unique existence using the coercivity of the bilinear form from the weak formulation. In this way, we could define the solution operator of the heat-convection phenomena and split it into the inhomogeneous part \( \hat{y} \) and the homogeneous part \( Su \) (see Definition and Remark 4.5).

Up next, we considered the optimal boundary control problem. Chapter 3 taught us how to construct the corresponding augmented Lagrange function. After some computations, we obtained the gradient of the augmented Lagrange function which includes the Hilbert adjoint of \( S \). In practice, we can not evaluate this adjoint. Therefore, we set up the adjoint equation (AE). The unique existence of the solution of this end value partial differential equation was provided by transforming it into an initial value problem. We defined the solution operators \( R_1 \) and \( R_2 \) in Definition 5.4 to get a representation of the solution of the adjoint equation. By doing this, we could rewrite the gradient of the augmented Lagrange function such that it was numerically practicable.

Regarding the numerical implementation, our first step was to solve the two arising partial differential equations with the finite element Galerkin ansatz combined with an implicit Euler scheme in time, i.e. we chose the linear finite element space and converted the partial differential equations into ordinary differential equations which were then treated by the implicit Euler. Inside the augmented Lagrangian algorithm, we minimized the Lagrange function. With having the gradient, we were able to apply a first-order method. In our case, it was the gradient projection method. The projecting function of this method also took on the task of the control constraints. An algorithm of solving the optimal control problem was created and tested in the last part of the thesis. We basically solved the same test than Luca Mechelli and Stefan Volkwein use in their paper (see [MV18]). By taking the very same functions and initial guesses, we obtained
similar results if we used a certain technique: The first iteration had to be done with a small penalty parameter $c$. For the next iteration, we were forced to push this parameter a lot with the increment $\beta$ while we had to decrease $\beta$ in the following loops and chose it within the common range $[4, 10]$ for Lagrangian methods (see [Ber99, page 405]). Without using this technique, the method could not converge to the results Mechelli and Volkwein obtained. However, in the second test, we only modified the initial guess for the control. This changed the execution of the augmented Lagrangian method. We were able to start with $c = 1$ and without the necessity of the technique above but using a common constant $\beta$, we obtained similar results. The previous tests concentrated on finding the so-called economic model predictive control (see [GP17, Chapter 8]) and in the last test, we actually tried to track the prescribed temperature profiles. In case we chose the right initial control, we obtained reliable results without the explained $c$-update. Summarizing, using the augmented Lagrangian method to solve our problem was efficient.

Nevertheless, there is still space for several further investigations: It is possible to apply a second-order method to minimize the augmented Lagrange function. Of course, the second derivative is then required. The question is whether the presumable lower number of iterations can save enough computing time such that it can top the extra time of calculating the Hessian matrix and solving the bigger system. Furthermore, utilizing the finite element method for optimal control problems can quickly demand high costs because finite element spaces are normally high-dimensional. Model order reduction may help to achieve a faster convergence. Its concept is to create a low-dimensional subspace which still owns the most central aspects of the investigated problem. We refer to the joint book [SvdVR08] from Schilders, van der Vorst and Rommes for an overview and exemplary applications. In particular, the specific method of proper orthogonal decomposition can be found there. In general, this method seems to be most widely used in optimal control of parabolic partial differential equations (see [TV09, pages 83-115], [ABK01, pages 1311-1330] and [AFS00, Report no. 2000-25] for instance).
REFERENCES


